

# **Preliminary Liquefaction Susceptibility Using Different Machine Learning Techniques**



**By**

**Sardar Obaidullah**

**NUST-2020-MS GEOTECH-327660**

**Department of Civil Engineering**

NUST Institute of Civil Engineering

School of Civil and Environmental Engineering

National University of Sciences & Technology (NUST)

Islamabad, Pakistan

2024

# **Preliminary Liquefaction Susceptibility Using Different Machine Learning Techniques**



By

Sardar Obaidullah

NUST-2020-MSGEO TECH-327660

A thesis submitted to the National University of Sciences and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Master of Science in Geotechnical Engineering

Supervisor: Dr. Abbas Haider

School of Civil and Environmental Engineering


National University of Sciences & Technology (NUST)

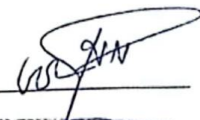
Islamabad, Pakistan

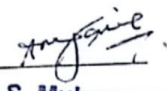
2024

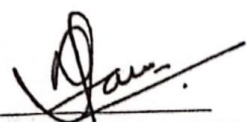
## THESIS ACCEPTANCE CERTIFICATE

It is certified that the final copy of the MS thesis written by Mr. Sardar Obaidullah, Registration No. 00000327660, of MS Geotechnical Engineering SCEE, (NICE), has been vetted by the undersigned, found completed in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS degree.

Signature:   
Supervisor: Dr. Abbas Haider  
Date: 30/04/2024

Signature (HoD):   
**HoD Geotechnical Engineering**  
NUST Institute of Civil Engineering  
School of Civil & Environmental Engineering  
Date: National University of Sciences and Technology

Signature (Associate Dean, NICE)   
**Dr. S. Muhammad Jamil**  
Associate Dean  
NICE, SCEE, NUST  
Date: 03/5/24

Signature (Principal & Dean):   
**PROF DR MUHAMMAD IRFAN**  
Principal & Dean  
SCEE, NUST  
Date: 06 MAY 2024


# National University of Sciences and Technology

## MASTER'S THESIS WORK

We hereby recommend that the dissertation prepared under our Supervision by **Sardar Obaidullah**, Registration No. **00000327660** Titled: **“Preliminary Liquefaction Susceptibility Using Different Machine Learning Techniques”** be accepted in partial fulfillment of the requirements for the award of a degree with (B+ Grade).

### Examination Committee Members

1. Name: Dr. Badee Alshameri

Signature: 

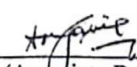
2. Name: Dr. Zain Maqsood

Signature: 

Supervisor's name: Dr. Abbas Haider

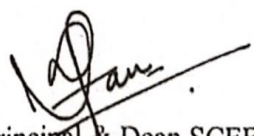
Signature: 

  
Head of Department  
HoD Geotechnical Engineering  
NUST Institute of Civil Engineering  
School of Civil & Environmental Engineering  
National University of Sciences and Technology

  
Dr. S. Muhammad Jamil  
Associate Dean  
NICE, SCEE, NUST  
(Associate Dean)

**COUNTERSIGNED**

Date: 08 MAY 2024

  
Principal & Dean SCEE  
PROF DR MUHAMMAD IRFAN  
Principal & Dean  
SCEE, NUST

CERTIFICATE OF APPROVAL

This is to certify that the research work presented in this thesis, entitled "Preliminary Liquefaction Susceptibility using Different Machine Learning Techniques" was conducted by Mr. Sardar Obaidullah under the supervision of Dr. Abbas Haider.

No part of this thesis has been submitted anywhere else for any degree. This thesis is submitted to the Nust Institute of Civil Engineering in partial fulfillment of the requirements for the degree of Masters in Geotechnical Engineering, NUST Islamabad.

Student Name: Sardar Obaidullah

Signature: 

Examination Committee:

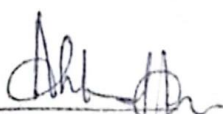
a) Dr. Badee Alshameri  
HOD Geotechnical Engineering  
(SCEE, NUST)

Signature: 

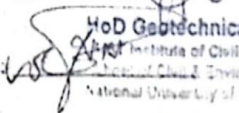
b) Dr. Zain Maqsood  
(Assistant Professor, SCEE, NUST)

Signature: 

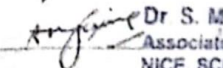
Dr. Abbas Haider  
(Supervisor)

Signature: 


Head of Department: Dr. Badee Alshameri

Signature:   
HOD Geotechnical Engineering  
Nust Institute of Civil Engineering  
Department of Civil & Environmental Engineering  
National University of Sciences and Technology

Associate Dean: Dr. Syed Muhammad Jamil

Signature:   
Dr. S. Muhammad Jamil  
Associate Dean  
NICE, SCEE, NUST

Principal & Dean: Dr. Muhammad Irfan

Signature:   
PROF. DR. MUHAMMAD IRFAN  
Principal & Dean  
SCEE, NUST

## **AUTHOR'S DECLARATION**

I Sardar Obaidullah hereby state that my MS thesis titled:

***Preliminary Liquefaction Susceptibility Using Different Machine Learning Techniques***

is my work and has not been submitted previously by me for taking any degree from the Nust Institute of Civil Engineering, Islamabad, or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after I graduate, the university has the right to withdraw my MS degree.

Name of Student: Sardar Obaidullah

Date: 30/04/2024

## **PLAGIARISM UNDERTAKING**

I solemnly declare that the research work presented in the thesis titled “**Prediction of Liquefaction Susceptibility using Different Machine Learning Techniques**” is solely my research work with no significant contribution from any other person. Small contribution/ help wherever taken has been duly acknowledged and I have written that complete thesis.

I understand the zero-tolerance policy of the HEC and NUST Institute of Civil Engineering towards plagiarism. Therefore, I as an author of the above-titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred to/cited.

I undertake that if I am found guilty of any formal plagiarism in the above-titled thesis even after the award of the MS degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized thesis.

Student/Author Signature: \_\_\_\_\_



Name: Sardar Obaidullah

## ACKNOWLEDGEMENT

I am extremely grateful to Almighty ALLAH, the most Merciful and Merciful, who provided me with knowledge and enlightenment to carry out this research endeavor. Countless salutations upon the Holy Prophet (P.B.U.H), the source of knowledge and guidance for mankind in every walk of life. I wish to express my sincere gratitude to my research supervisor Dr. Abbas Haider who continuously and convincingly conveyed a spirit of hardworking and steadfastness to contribute and complete this project. Without his tireless efforts, support, and guidance, the completion of this project would not have been possible.

I would also like to express my gratitude to Dr. Badee Alshameri and Dr. Zain Maqsood. Their eagerness for my research work and their invaluable efforts towards the completion of my thesis work cannot be disregarded.

I am extremely indebted to Dr. Badee Alshameri for all the inspiration and guidance I obtained from him during my studies. I am deeply grateful to my parents for their love, support, and hard work, thanking them for their endless patience and encouragement when it was most needed.



## ABSTRACT

Liquefaction Analysis is one of the most important parameters in the study of geotechnical earthquake engineering. Till the 1960's this phenomenon was relatively less unearthed, and practitioners have not formulated any detailed method for its assessment. However, two large earthquakes in 1964 in Niigata, Japan, and Alaska, USA have turned the attention of engineers toward this issue. Several laboratory and field methods were developed for the calculation of the Factor of Safety. Developed methods have been used all over the world which use the Cyclic Resistance Ratio and Cyclic Stress Ratio for measuring the factor of safety. As empirical methods have inherent limitations due to certain assumptions for the ease of work, liquefaction formulations have also been generalized to counter the real-world scenario. These assumptions significantly impacted the implementation of the simplified methods. In recent years Machine Learning has been used to improve the inherent shortcomings present in the conventional method of predicting liquefaction. As machine learning algorithms learn from the data and are not explicitly programmed, they can develop highly non-linear relationships and learn from the data. Researchers have mostly used the published data based on the Factor of Safety to predict the liquefaction potential which is not a credible approach as it has inherent shortcomings in ascertaining CSR and CRR(Kurnaz & Kaya, 2019a).In this research work four different machine learning models namely Logistic regression, Support vector machine, Decision tree, and Artificial neural networks have been used to predict the liquefaction potential of the soil based on the published field data. The data has been procured from credible published research papers. It includes six different input parameters named cone tip resistance, sleeve friction ratio, effective stress, total stress, maximum horizontal ground surface acceleration, earthquake moment magnitude, and one output parameter named liquefaction. The performance of the developed models was gauged with the help of classification assessment report parameters named accuracy, precision, recall, and F1 score. It was found that the Decision Tree algorithm performance

was the best among all the other algorithms followed by Artificial neural networks, Logistic regression, and Support vector machine. The developed models can be used as a predictive model for the preliminary liquefaction assessment of soil.

**Keywords:** Liquefaction, Machine Learning, Cyclic Resistance Ratio (CRR), Cyclic Stress Ratio, (CSR), Decision Tree, Logistic Regression, Support Vector Machine, Artificial Neural Network.

## TABLE OF CONTENTS

|  | <b>Page No.</b> |
|--|-----------------|
| <b>ACKNOWLEDGEMENT</b> .....   | <b>viii</b>     |
| <b>ABSTRACT</b> .....  | <b>ix</b>       |
| <b>TABLE OF CONTENTS</b> .....   | <b>xi</b>       |
| <b>LIST OF FIGURES</b> .....   | <b>xiii</b>     |
| <b>LIST OF TABLES</b> .....  | <b>xv</b>       |
| <b>LIST OF EQUATIONS</b> .....   | <b>xvi</b>      |
| <b>NOTATIONS</b> .....   | <b>xvii</b>     |
| <b>1. INTRODUCTION</b> .....   | <b>1</b>        |
| <b>1.1. General</b> .....  | <b>1</b>        |
| <b>1.2. Fundamentals of Liquefaction</b> .....   | <b>1</b>        |
| <b>1.3. Liquefaction Susceptibility</b> .....  | <b>2</b>        |
| <i>1.3.1. Historical Criteria; Geological Criteria; and Composition Criteria</i> ..... | <i>3</i>        |
| <i>1.3.2. State Criteria</i> .....   | <i>4</i>        |
| <b>1.4. Liquefaction Initiation</b> .....  | <b>7</b>        |
| <i>1.4.1. Flow Liquefaction</i> .....  | <i>7</i>        |
| <i>1.4.2. Cyclic Mobility</i> .....  | <i>8</i>        |
| <b>1.5 Factor of Safety for Determining Liquefaction Potential</b> .....               | <b>10</b>       |
| <i>1.5.1 Simplified Method (Originally Seed and Idriss 1971):</i> .....                | <i>10</i>       |
| <i>1.5.2 Inherent Shortcomings to Compute CSR by Simplified Method:</i> .....          | <i>11</i>       |
| <i>1.5.3 Determination of CRR:</i> .....   | <i>11</i>       |
| <b>1.6 Problem Statement</b> .....   | <b>13</b>       |
| <b>1.7 Machine Learning as an Alternative</b> .....                                    | <b>14</b>       |
| <b>2. LITERATURE REVIEW</b> .....  | <b>15</b>       |
| <b>3. METHODOLOGY AND RESEARCH WORK</b> .....  | <b>19</b>       |
| <b>3.1 Overview of the Machine Learning Algorithms Used</b> .....                      | <b>19</b>       |
| <i>3.1.1 Support Vector Machine</i> .....  | <i>19</i>       |
| <i>3.1.2 Logistic Regression</i> .....   | <i>21</i>       |
| <i>3.1.3 Decision Trees</i> .....  | <i>24</i>       |
| <i>3.1.4 Artificial Neural Network:</i> .....  | <i>26</i>       |
| <b>4. DATASET DESCRIPTION AND EXPLORATORY DATA ANALYSIS</b> .....                      | <b>29</b>       |
| <b>4.1 Dataset Description:</b> .....  | <b>29</b>       |

|   |           |
|---|-----------|
| <b>4.2 Statistical Description of Data</b> .....            | <b>30</b> |
| 4.2.1 <i>Correlation Matrix</i> .....                       | 30        |
| <b>4.3 Exploratory Data Analysis</b> .....                  | <b>33</b> |
| 4.3.1 <i>Histogram</i> .....                                | 33        |
| 4.3.2 <i>Joint plots</i> .....                              | 35        |
| 4.3.3 <i>Box Plots</i> .....                                | 36        |
| 4.3.4 <i>Swarm Plot</i> .....                               | 40        |
| 4.3.5 <i>Pair plot</i> .....                                | 41        |
| <b>5. PERFORMANCE EVALUATING PARAMETERS</b> .....           | <b>42</b> |
| <b>5.1 Confusion Matrix</b> .....                           | <b>42</b> |
| <b>5.2 Accuracy</b> .....                                   | <b>43</b> |
| <b>5.3 Precision</b> .....                                  | <b>44</b> |
| <b>5.4 Recall</b> .....                                     | <b>44</b> |
| <b>5.5 F1 Score</b> .....                                   | <b>44</b> |
| <b>6. PYTHON PROGRAMMING, RESULTS AND DISCUSSIONS</b> ..... | <b>45</b> |
| <b>6.1 Python Programming</b> .....                         | <b>45</b> |
| 6.1.1 <i>Logistic Regression Classification</i> .....       | 45        |
| 6.1.2 <i>Logistic Regression Results</i> .....              | 47        |
| 6.1.3 <i>Decision Tree Classification</i> .....             | 48        |
| 6.1.4 <i>Decision Tree Results</i> .....                    | 50        |
| 6.1.5 <i>Support Vector Machine Classification</i> .....    | 51        |
| 6.1.6 <i>Support Vector Machine Results</i> .....           | 52        |
| 6.1.7 <i>Artificial Neural Network Classification</i> ..... | 53        |
| 6.1.8 <i>Artificial Neural Network Results</i> .....        | 55        |
| <b>6.2 Discussion</b> .....                                 | <b>56</b> |
| <b>6.3 Practical Demonstration of the Algorithm</b> .....   | <b>60</b> |
| <b>7. CONCLUSION AND FUTURE RECOMMENDATIONS</b> .....       | <b>63</b> |
| <b>7.1 Conclusion</b> .....                                 | <b>63</b> |
| <b>7.2 Recommendations</b> .....                            | <b>65</b> |
| <b>References</b> .....                                     | <b>66</b> |

## LIST OF FIGURES

|   | <b>Page No.</b> |
|---|-----------------|
| figure 1 Casagrande’s Cvr Line (After Kramer; 1996).....          | 5               |
| Figure 2 Castro’s Triaxial Tests (After Kramer, 1996) .....       | 5               |
| Figure 3 Castro's Steady State Line .....                         | 6               |
| Figure 4 Flow Liquefaction Surface .....                          | 8               |
| Figure 5 Space (In P’ – Q Graph) Prone To Flow Liquefaction ..... | 8               |
| Figure 6 Space (In P’-Q Graph) Prone To Cyclic Mobility .....     | 9               |
| Figure 7 Cyclic Mobility Scenarios .....                          | 9               |
| Figure 8 Depth Reduction Factor Graph.....                        | 11              |
| Figure 9 Csr Vs N1(60) Graph By Seed Et Al.....                   | 13              |
| Figure 10 Svm Algorithm Graphic Intuition.....                    | 19              |
| Figure 11 Kernel Function For Non-Linearity.....                  | 20              |
| Figure 12 Logistic Regression For Binary Classification.....      | 22              |
| Figure 13 Graphical Intuition Of Logistic Regression .....        | 23              |
| Figure 14 Decision Tree Model Intuition.....                      | 25              |
| Figure 15 Artificial Neural Network Structure .....               | 28              |
| Figure 16 Data Head In Jupyter Notebook .....                     | 29              |
| Figure 17 Statistical Description Of Input Data .....             | 30              |
| Figure 18 Correlation Matrix Of The Data Set .....                | 31              |
| Figure 19 Heatmap Of The Correlation Matrix .....                 | 32              |
| Figure 20 Bar Graph Of The Correlation Matrix.....                | 32              |
| Figure 21 Tip Resistance Histogram.....                           | 33              |
| Figure 22 Sleeve Friction Histogram.....                          | 34              |
| Figure 23 Effective Stress Histogram .....                        | 34              |
| Figure 24 Joint Plot Of Total Stress And Cone Resistance.....     | 35              |
| Figure 25 Joint Plot Of Sleeve Friction And Cone Resistance ..... | 36              |
| Figure 26 Boxplot Of Input Variables.....                         | 36              |
| Figure 27 Tip Resistance Box Plot.....                            | 37              |
| Figure 28 Outliers Removed Tip Resistance Box.....                | 38              |
| Figure 29 Total Stress Box Plot .....                             | 39              |
| Figure 30 Outliers Removed Total Stress Box Plot .....            | 39              |
| Figure 31 Swarm Plot Of The Data Points .....                     | 40              |
| Figure 32 Pair Plot Of The Datasets .....                         | 41              |

Figure 33 Confusion Matrix Binary Classification..... 43  
Figure 34 Python Code Of Logistic Regression ..... 46  
Figure 35 Confusion Matrix Of Logistic Regression ..... 47  
Figure 36 Python Code Of Decision Tree Classification..... 49  
Figure 37 Decision Tree Confusion Matrix ..... 50  
Figure 38 Support Vector Machine Python Code ..... 51  
Figure 39 Support Vector Machine Confusion Matrix ..... 52  
Figure 40 Artificial Neural Network Python Code ..... 54  
Figure 41 Confusion Matrix Of Ann..... 55  
Figure 42 Summary Report Of All Algorithms Used ..... 58

## LIST OF TABLES

|   | <b>Page No.</b> |
|---|-----------------|
| Table 1 Logistic Regression Graphical Performance.....        | 48              |
| Table 2 Tree Classifier Graphical Performance .....           | 50              |
| Table 3 Support Vector Machine Graphical Performance .....    | 52              |
| Table 4 Artificial Neural Network Graphical Performance ..... | 55              |
| Table 5 Logistic Regression Report.....                       | 56              |
| Table 6 Support Vector Machine Report.....                    | 56              |
| Table 7 Artificial Neural Network Report .....                | 57              |
| Table 8 Decision Tree Report .....                            | 57              |
| Table 9 Sensitivity Analysis.....                             | 59              |
| Table 10 Test Data With Tip Resistance 8 Mpa .....            | 60              |
| Table 11 Test Data With Tip Resistance 6 Mpa .....            | 61              |
| Table 12 Test Data With Tip Resistance 4 Mpa .....            | 62              |

## LIST OF EQUATIONS

|                   | <b>Page No.</b> |
|-------------------|-----------------|
| Equation 1 .....  | 10              |
| Equation 2 .....  | 10              |
| Equation 3 .....  | 13              |
| Equation 4 .....  | 22              |
| Equation 5 .....  | 22              |
| Equation 6 .....  | 22              |
| Equation 7 .....  | 25              |
| Equation 8 .....  | 26              |
| Equation 9 .....  | 27              |
| Equation 10 ..... | 27              |
| Equation 11 ..... | 43              |
| Equation 12 ..... | 44              |
| Equation 13 ..... | 44              |
| Equation 14 ..... | 44              |



## NOTATIONS

|          |                           |
|----------|---------------------------|
| ML.....  | Machine Learning          |
| AI.....  | Artificial Intelligence   |
| ANN..... | Artificial Neural Network |
| LOG..... | Logistic Regression       |
| SVM..... | Support Vector Machine    |
| DT.....  | Decision Tree             |
| CSR..... | Cyclic Stress Ratio       |
| CRR..... | Cyclic Resistance Ratio   |
| LL.....  | Liquid Limit              |
| PL.....  | Plastic Limit             |
| DL.....  | Deep Learning             |

# ***Chapter. 1***

## **1. INTRODUCTION**

### **1.1. General**

The liquefaction of soil presents a significant challenge in geotechnical engineering, particularly in regions prone to seismic activity. This phenomenon occurs when saturated soil loses its strength and stiffness under the influence of cyclic loading, such as earthquakes, resulting in a temporary state resembling that of a liquid. Liquefaction can lead to catastrophic consequences, including ground settlement, structural damage, and even the collapse of buildings and infrastructure.

The liquefaction of soil presents a significant challenge in geotechnical engineering, particularly in regions prone to seismic activity. This phenomenon occurs when saturated soil loses its strength and stiffness under the influence of cyclic loading, such as earthquakes, resulting in a temporary state resembling that of a liquid. Liquefaction can lead to catastrophic consequences, including ground settlement, structural damage, and even the collapse of buildings and infrastructure.

Addressing the liquefaction susceptibility of soil requires a comprehensive understanding of various factors such as soil composition, groundwater levels, seismic characteristics, and historical earthquake data. Traditional geotechnical methods for assessing liquefaction potential involve extensive field investigations and laboratory testing, which can be time-consuming and costly.

### **1.2. Fundamentals of Liquefaction**

"Liquefaction" is a term used to describe various phenomena; the common thing allied with all phenomena is the instigation of excess water pressure due to the increase in the dynamic load. (Kramer & Seed, 1988)

During the process of un-drained loading, the capacity of saturated soil to contract initiates the development of excess soil pore water pressure in the soil matrix, which results in a decrease in the effective stress of the soil matrix. The reduction in effective stress further leads to various failures. The aforementioned failures are classified into two main types: **flow liquefaction and cyclic mobility**. Cyclic mobility failure happens more regularly and often causes less damage than flow liquefaction.

*Flow liquefaction* is most commonly described by substantial, abrupt deformations influencing large areas (Kramer, 1996). Flow liquefaction initiates when the static shear stresses of a soil mass exceed the soil's inherent shear strength. The precarious state results in an abrupt movement of the matrix, called flow failure. In flow failures, the static stresses-which are shear stresses in nature are greater than the soil's strength.

Cyclic mobility, in contrast to flow-liquefaction, starts when static shear stress does not exceed the shear strength of the soil mass(Kramer & Seed, 1988).

Three topics must be thoroughly understood to address hazards associated with the liquefaction phenomenon: Liquefaction Susceptibility, Liquefaction Initiation, and Liquefaction Effects. Soil might not be prone to liquefaction at the very start. Even if a soil mass is vulnerable to liquefaction, the prevalent environment is not there to start the process of liquefaction. Finally, if liquefaction initiates, adverse impacts may not be exhibited.

### **1.3. Liquefaction Susceptibility**

The inquiry if a soil is at risk of liquefaction hazard or not is the fundamental step towards the determination of liquefaction potential. Soil vulnerability to liquefaction is determined by the use of different criteria which include and are not limited to historical, geological, and composition, and state criteria.

### *1.3.1. Historical Criteria; Geological Criteria; and Composition Criteria*

Historical criteria are rudimentary and basic considerations of liquefaction; questioning whether soil is liquefied or not in its history. This factor is mostly premised on different case histories or on-site real proof of earlier liquefaction. (Rinne, 1987) ascertained that as long as soil and groundwater conditions do not vacillate, liquefaction occurs at the same location.

Geological criteria consider geological footprints in the evaluation of a soil's vulnerability to liquefaction (Youd & Keefer, 1994).

Conventionally, liquefaction was taken into matter just in wet soils (Kramer & Seed, 1988); and ground settings primarily controlled whether a soil is at liquefaction risk or not. However, tri-axial tests conducted by (Unno et al., 2008) highlighted that certain unsaturated soils also drop effective stress due to the cyclic shear and act as liquids. They detected this phenomenon to initiate when the air in the pores of the soil matrix and the pressure of water is equal to the starting confining pressure. Therefore, unsaturated soils should not be taken out of the equation while assessing the liquefaction potential.

Composition criteria have been altered significantly in recent years. Many practitioners, for almost a decade, trusted the "Chinese criteria" (Liu, 2020) which says that soils can be prone to liquefaction hazards if the given conditions are satisfied:

- Fraction which is finer than 0.005mm  $\leq$  15%
- Liquidity Index of the soil  $\leq$  0.75
- Water Content of the soil,  $w_c \geq 0.9$  LL

- Liquid Limit,  $LL \leq 35\%$

This rudimentary criterion was pervasive until several profound earthquakes (1994 Northridge Earthquake, Earthquake of Kocaeli 1999, and Chi-Chi 1999 Earthquake) in which a large amount of infrastructure was damaged due to the Chinese criteria.

### 1.3.2. State Criteria

Engineers would analyze the soil and ask a few questions; is this soil susceptible to liquefaction, and would look at the historical, geologic, and compositional criteria to try to answer the question. If the answer to the question is a yes, that soil is vulnerable to liquefaction, it does not necessarily mean that the soil will liquefy for sure if it is exposed to earthquakes. It is because the initiation also depends upon the magnitude and duration of the loading that we are applying. Casagrande (1936) established primitive patterns to assess the liquefaction potential. He performed drained, tri-axial tests on contractive and dilative sands and observed all soils regardless of dense or loose, when they are sheared, soils try to approach a line which is called the Critical Void Ratio Line. The void ratio to where all soils come together was named the *critical void ratio*  $e_c$ , Casagrande speculated that the undrained shearing of loose matrix specimens produces pore pressure positive in nature and on the contrary dense specimens produce negative pore water pressure. According to Casagrande, this line would be the margin connecting the susceptible and unsusceptible soils. Soils present above the Critical Void Ratio line as shown in Figure 1.1 were vulnerable to liquefaction, and soils present below the CVR line were dense sands, and hence resistant to liquefaction.

Understanding these criteria is crucial for assessing and mitigating the risk of soil liquefaction, especially in areas prone to seismic activity or other cyclic loading events. Engineering measures, such as soil improvement techniques and proper foundation design, can help mitigate the risks associated with liquefaction.

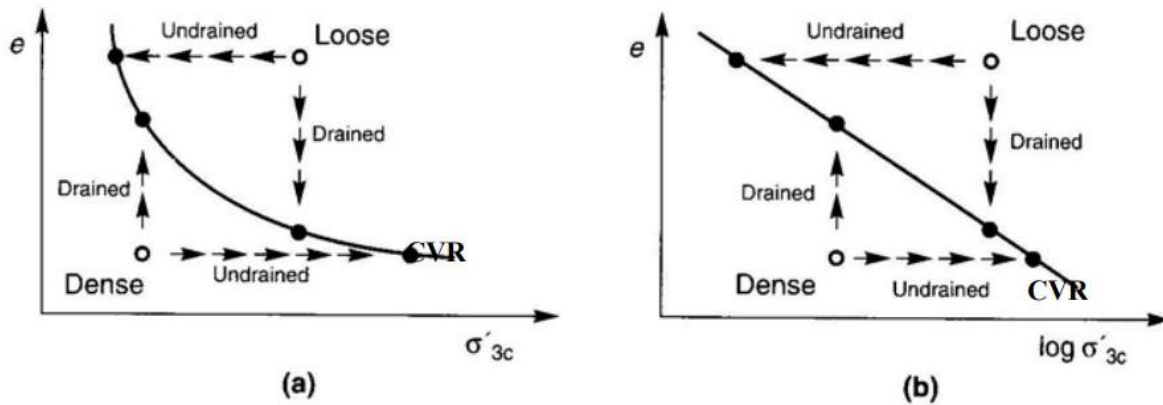


Figure 1 Casagrande's CVR Line (after Kramer; 1996)

The explanation given by Casagrande was considered to be rational and true until Fort Peck Dam failed in 1938. Post-dam failure investigations of the site divulged that the initial condition of almost all the liquefied soils was beneath the Critical Void Ratio Line. To rectify the mistakes of experiments, Castro, conducted several cyclic tests on isotopically consolidated specimens and developed a concept slightly different from Casagrande's assumption (Castro, 1969). He demonstrated that the soil specimens acted in three different ways. According to him loose soil specimens in (Figure 1.2) show contraction under loading and tend to liquefy under these loads.

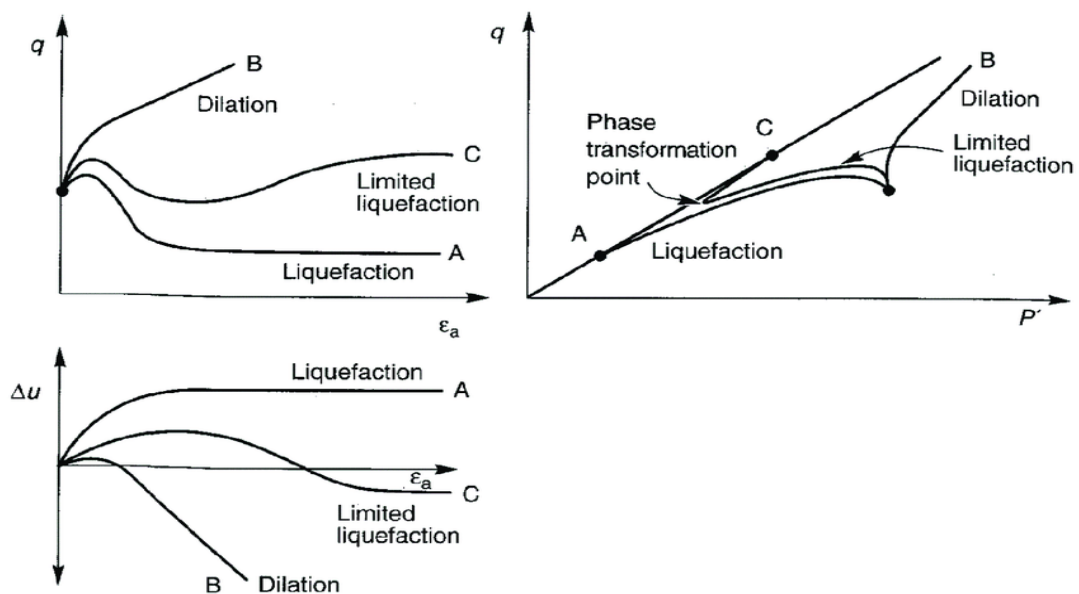


Figure 2 Castro's Triaxial Tests (after Kramer, 1996)

On the other hand, very dense specimens (Specimen B) show initial contraction but later dilate and grow strength; no tangible proof of the liquefaction is visible. Castro's research had given rise to the phenomenon of the steady state of deformation. Steady-state deformation of soil in liquefaction refers to the continuous and ongoing displacement or movement of soil particles following the onset of liquefaction. When saturated soils experience cyclic loading, such as during an earthquake, the pore water pressure within the soil increases, leading to a reduction in effective stress and subsequent loss of shear strength. This decrease in strength can cause the soil to transition from a solid-like state to a liquid-like state, resulting in flow-like deformation. (Poulos et al., 1985) defined it as a state where soil continuously deformed under persistent shear stress and confining effective pressure. This soil state is known as the steady state strength,  $S_{su}$ . The *steady-state line* runs parallel to the CVR, but slightly below it. It represents a true boundary between dilative and contractive behavior in undrained conditions.

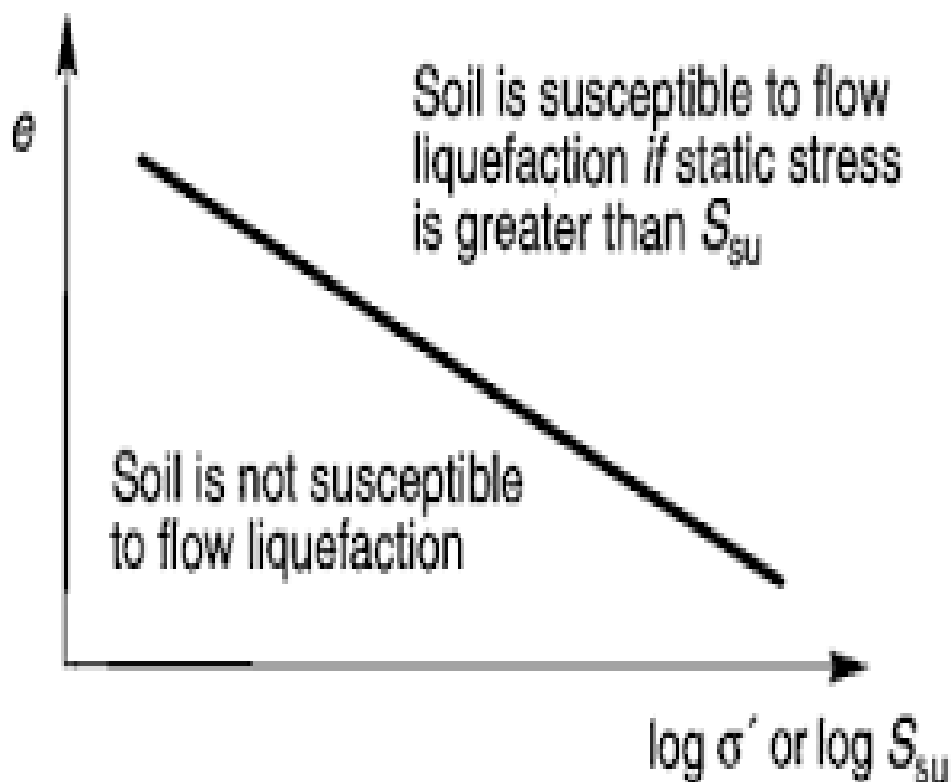


Figure 3 Castro's Steady State Line

Castro's work was so indispensable because now engineers were able to show that if the soil is tested in an undrained manner, it almost shifts the CVR line, and we use a steady state line to tell whether the soil is contractive or in other words susceptible to liquefaction.

## 1.4. Liquefaction Initiation

Liquefaction initiation is studied in two classes.

- Flow Liquefaction
- Cyclic Mobility

### 1.4.1. Flow Liquefaction

In 1979 (Hanzawa et al) elucidated that liquefaction initiation could be explained with the use of stress paths of loose saturated soils. In Figure 1.4, there are five specimens consolidated at the same void ratio showing different behaviors at different initial confining stress (undrained tests). A and B soil specimens are below the steady state line and demonstrate dilative behavior as the stress paths go to the steady state point. These specimens show no signs of liquefaction. On the contrary, C, D, and E soil specimens are above the line and show contractive behavior. These specimens experience a peak shear strength before reaching the ssp (steady state point), and this peak strength is the point where specimens start to lose the capability to take any load. (Vaid et al., 1985) propounded a line of peak strengths known as *flow liquefaction surface* (FLS). It is a line or surface where if the stress path hits it, flow liquefaction is initiated, and the stress path rapidly dives to the steady state strength. FLS truncates at a certain position; it is because flow liquefaction cannot take place if the initial state of stress of soil is less than the steady state point. (Galupino & Dungca).



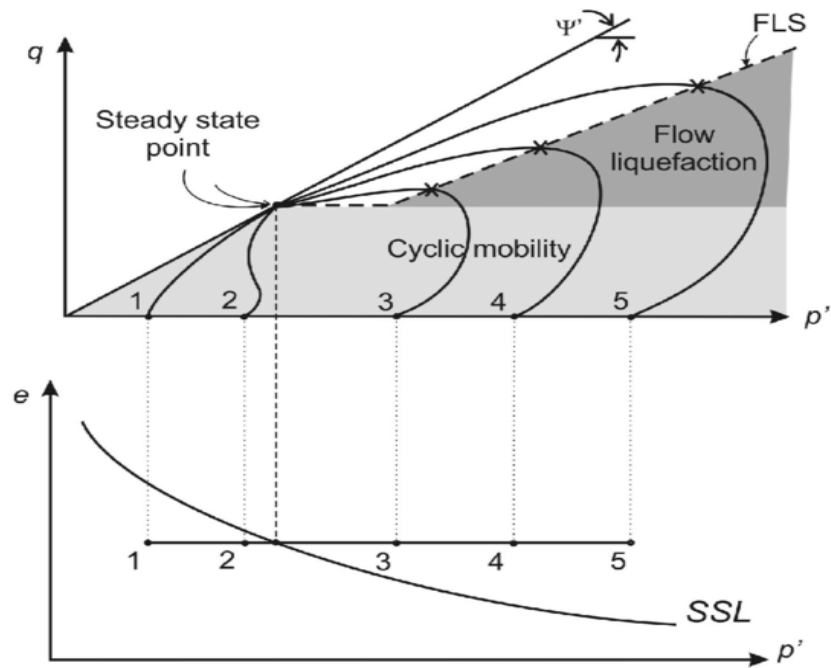


Figure 4 Flow Liquefaction Surface

If a soil specimen is in the mentioned area, and experiences an undrained loading, there is a high risk that the flow liquefaction would begin.

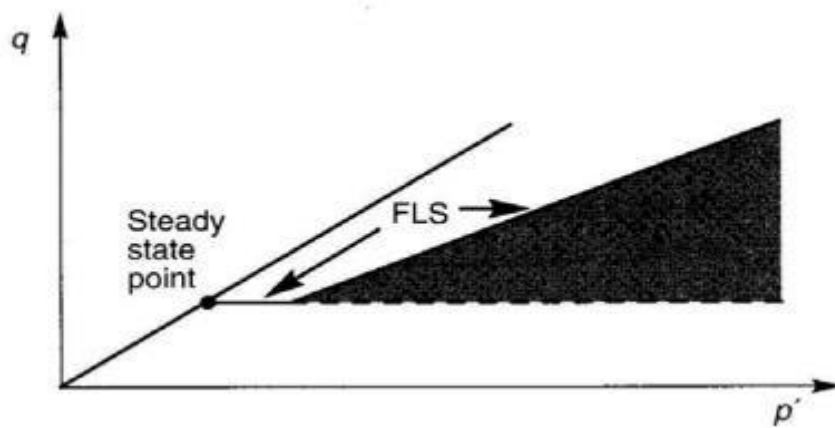


Figure 5 Space (in  $p' - q$  graph) Prone to Flow Liquefaction

#### 1.4.2. Cyclic Mobility

Cyclic mobility in soil liquefaction refers to the phenomenon where soils lose their strength and stiffness temporarily during cyclic loading, such as earthquakes or other repetitive loading events. This process can cause the soil to behave like a liquid, resulting in potentially hazardous consequences for structures built upon it. During seismic activity or cyclic loading, the soil

experiences stress from the shaking or loading. This stress causes an increase in pore water pressure within the soil mass, which, when it surpasses the effective stress, reduces the effective stress to zero. As a result, the soil loses its strength and behaves like a viscous liquid, potentially leading to settlement, lateral spreading, and even structural failure.

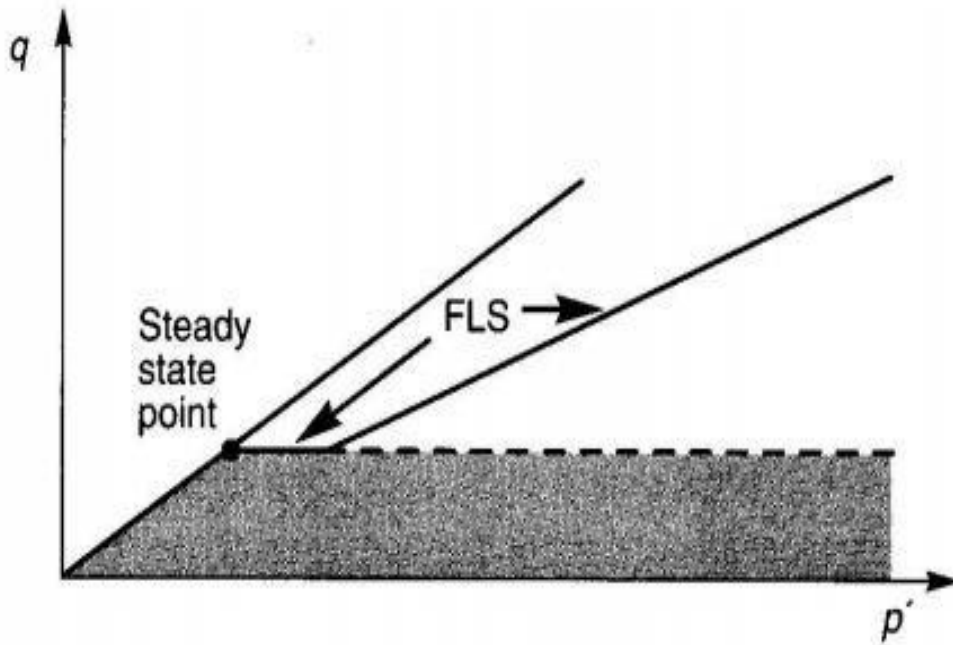


Figure 6 Space (in  $p'$ - $q$  graph) Prone to Cyclic Mobility

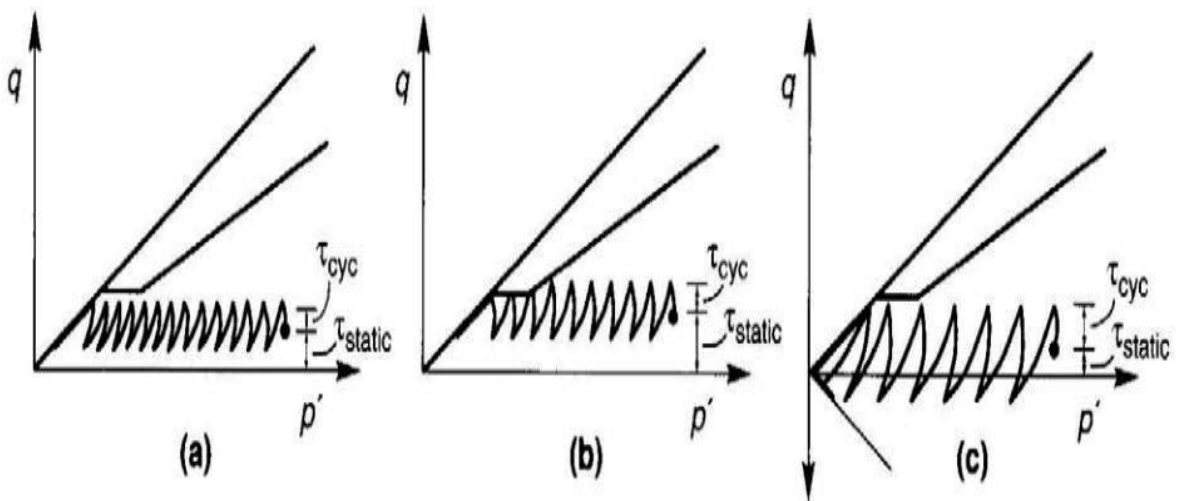


Figure 7 Cyclic Mobility Scenarios

## 1.5 Factor of Safety for Determining Liquefaction Potential

The potential for liquefaction initiation has been determined since the very beginning in terms of capacity, demand, and factor of safety.

$$FS_{Liq} = \frac{\text{capacity}}{\text{demand}} = \frac{\text{resistance}}{\text{loading}} = \frac{\tau_{cyc,L}}{\tau_{cyc}}$$

$FS_{Liq}$  is what is the shear stress required to liquefy the soil versus what is the shear stress applied to the soil. If the term is divided by effective overburden pressure, it would transform into a unitless equation containing Cyclic Resistance Ratio and Cyclic Stress Ratio.

$$FOS_{Liq} = \frac{\tau_{cyc,L}/\sigma_v'}{\tau_{cyc}/\sigma_v'} = CSR/CRR \quad (1)$$

- CRR quantifies the resistance of soil to liquefaction.
- CSR quantifies the cyclic loading from a particular earthquake.

CSR is computed these times using one of the two methods

- Site Specific Site Response Analysis
- Simplified Method

### 1.5.1 Simplified Method (Originally Seed and Idriss 1971):

Most engineers use this approach to approximate CSR.

$$(\tau_{cyc})_{rigid} = 0.65 \frac{a_{max}}{g} \sigma_v \quad (2)$$

Where;

$a_{max}$  = maximum horizontal ground acceleration

$g$  = gravitational acceleration

$\sigma_v$  = overburden vertical stress

( $T_{max}$ ) has been reduced to account for the fact that field conditions do not apply harmonic loading resulting in lower shear stresses. However, the soil is flexible, and not rigid. This is corrected by the *depth reduction factor*.

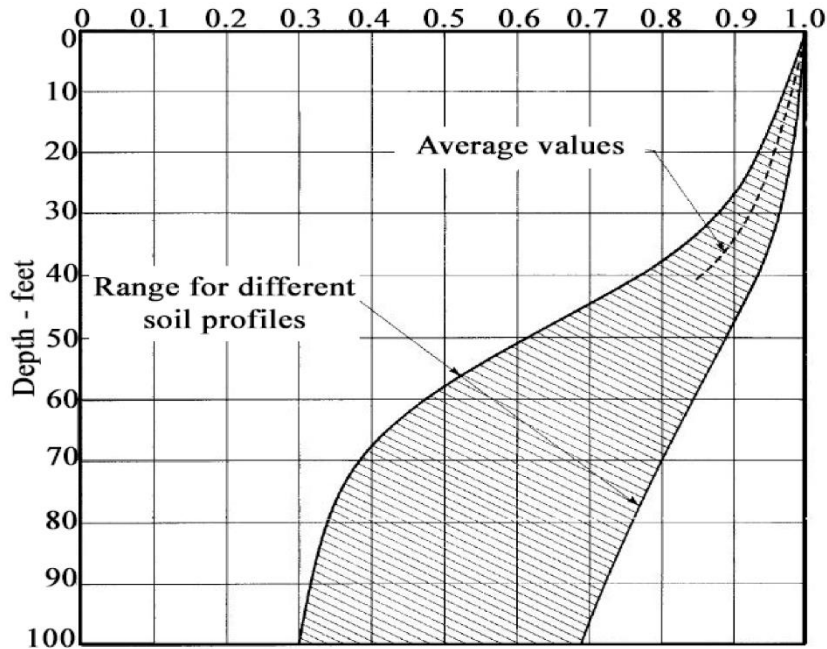


Figure 8 Depth Reduction Factor Graph

#### 1.5.2 Inherent Shortcomings to Compute CSR by Simplified Method:

- Pattern of Loading is harmonic which is extremely different from the site.
- Block of soil is considered to be rigid, which is also not the case in the field.

#### 1.5.3 Determination of CRR:

The Cyclic Resistance Ratio is determined by one of the two methods, by:

- Laboratory Testing
- Insitu Field Tests

#### ➤ Laboratory Testing:

Japan was prominent in Liquefaction research at the same time the USA was conducting several tests for the liquefaction prediction. Both countries in the 1960's relied unequivocally on Laboratory testing to investigate the triggering of liquefaction initiation, and both agreed that laboratory testing is indispensable; because stresses produced on the soil specimens and the

number of cycles of those stresses can be controlled. In this way, engineers can see if a particular soil will liquefy under a certain cyclic stress shear stress.

However, there were two major problems.

- How to get an undisturbed sample of the soil?
- How to reduce the costs of the tests?

Every time engineers try to sample sand; it rearranges its soil matrix. The only way to do so is to freeze the soil and core it. After these procedures, the soil could be transported. It takes a lot of time and money.

Initially, all the engineers preferred laboratory testing for CRR. However, in the USA, *Late Professor Harry Seed* thought if this practice continued, it would bankrupt the projects. Taking undisturbed soil samples is arduous, as the sand matrix rearranges every time it is disturbed. Engineers in Japan used the freezing method for the undisturbed soil sample which is way more costly.

#### *1.5.3.1 Main Idea by Prof. Harry B. Seed:*

Professor Seed's notable contribution is the development of liquefaction susceptibility criteria, such as the widely used "Seed and Idriss" method, which helps engineers assess the potential for soil liquefaction at a given site based on factors like soil properties, earthquake characteristics, and groundwater conditions. The main idea propounded by Professor Harry B. Seed is:

- Perform the SPT test and measure the blow count.
- Determine CSR from the design Earthquake.
- Plot CSR vs  $(N_1)_{60}$
- Plot the point with field SPT, if it plots above the CRR line, it will liquefy, otherwise it will not.

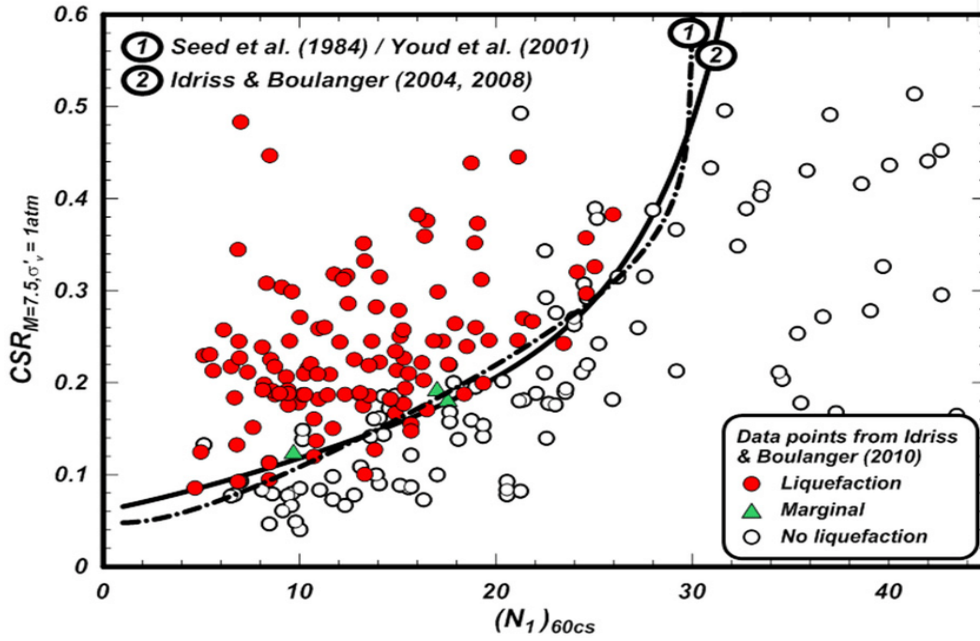


Figure 9 CSR vs  $N_1(60)$  Graph by Seed et al

$$\frac{\tau_{cyc,L}}{\sigma_v'} = CRR_{M=7.5, \sigma v'=1 atm} MSF \cdot K\sigma \cdot K\alpha \quad (3)$$

MSF= Magnitude Scaling Factor

$K\sigma$  = Overburden Correction Factor

$K\alpha$  = Initial Shear Stress Correction Factor

### 1.5.3.2 Inherent Shortcomings to Compute CRR:

The initial Shear Stress Correction Factor which assumes that the ground is leveled and there are no initial shear stresses present on the surface does not reflect the site conditions as the soil site is never leveled as propounded in the formulation of the formula.

## 1.6 Problem Statement

The methods and techniques used hitherto in estimating Liquefaction Susceptibility were premised on the Factor of Safety Method, which has inherent shortcomings in terms of estimating

the Cyclic Resistance Ratio and Cyclic Stress Ratio. New ways should be introduced to estimate the Liquefaction Susceptibility that can properly model the field and soil behavior under dynamic loading.

### **1.7 Machine Learning as an Alternative**

As has been explained there are multiple shortcomings in the determination of Factor of Safety for Liquefaction. Many researchers have used different field tests to counter the limitations; however, all these procedures have limitations for the computation. Recently, researchers have applied machine learning techniques to compute the liquefaction phenomenon as it is an intelligent computer-based model to compute the non-linear relationship among different variables. They found out that intelligent machine learning models have performed better than the conventional approach. This, of course, is due to the limitations that conventional methods have in terms of certain assumptions and lack of expertise in performing laboratory tests. In this research, different supervised machine learning algorithms such as Decision Tree, Logistic Regression, Support Vector Machine, and Artificial Neural Networks will be used to predict the liquefaction potential of the soil.

## ***Chapter. 2***

### **2. LITERATURE REVIEW**

The advancement in the field of artificial intelligence has allowed researchers to bring it into the realm of geotechnical engineering. (Jian, Xibing, & Xiuzhi, 2012) (Samui et al., 2011) (Hanna et al., 2007)(Khandelwal et al., 2018)(Zhou et al., 2015)(SHI et al., 2012). (Artificial Neural Network) is a deep learning method in machine learning, which can perform functions simulating the human mind and has been widely used for predicting liquefaction events. (Hanna et al., 2007)(Abbaszadeh Shahri, 2016)(Ramakrishnan et al., 2008).

(Zhou et al., 2022) used Genetic Algorithms along with Support Vector Machine to forecast the earthquake-induced liquefaction potential of the soil. A multi-data set was employed to develop the machine learning algorithms. As SVMs are sensitive to noise and outliers in the data, outliers can significantly affect the decision boundary, leading to poor performance, especially in high-dimensional spaces. This research aims to not only detect the outliers but also remove them to develop a machine-learning model for the prediction. (J. Zhang & Wang, 2021) used an Ensemble learning algorithm to predict the liquefaction assessment by the use of a Voting Classifier. Different base models were used to predict liquefaction prediction. This research used Cyclic Stress Ratio as a base input parameter for the model generation, which means dataset accuracy was compromised as CSR is determined using empirical relationship, and its accuracy is contested. (Kumar et al., 2021) predicted a deep machine learning (DL) model for classifying the soil in determining the liquefaction. Emotional Back Propagation neural networks were used to test the applicability of the model. In this research only two input parameters were used: Cone Penetration Test and Peak Ground Acceleration. When there are too few



input variables, the model may not have enough information to accurately identify the underlying patterns in the data. Consequently, it may resort to memorizing the training data rather than learning meaningful relationships, leading to overfitting.

(Ahmad et al., 2021) examined the implementation of different machine learning (ML) algorithms by (CPT) test based on case histories to ascertain the earthquake-induced liquefaction potential. An insufficient data set was used to train the model. Also, data was not balanced between liquefaction and non-liquefaction cases which misled the accuracy parameters. The results presented in the research did not determine the probability or the extent to which liquefaction is susceptible. This research work will use a probabilistic method to ascertain the likelihood of liquefaction potential. Similarly, (W. Zhang & Goh, 2018) assessed the liquefaction potential of soils based on the backpropagation networks. Also, they validated the model accuracy, F1 score, and AOC curve with the already published experimental data and found that backpropagation neural networks perform better than the simplified procedure equations. The model generated could not be applied to the dataset outside the range given by the author which limits its applicability.

(Zhou et al., 2019) proposed the stochastic-gradient-boosting (SGB) method for predicting soil liquefaction potential using SPT and CPT data history cases. These techniques have achieved good and confirming results. However, it has some limitations which include black-box nature. It overfits the algorithm and has a slow convergence speed. To counter these problems, (Kurnaz & Kaya, 2019b) proposed an alternative approach that uses the group method of data handling (GMDH). GMDH is a self-establishing machine-learning approach. Recently, this method has been applied to geotechnical problems. However, ensemble learning models are computationally expensive and time-consuming. They also overfit the problem's data.

In 2018 (Hoang & Bui) proposed an algorithm premised on the hybridization of the kernel discriminate investigation Support Vector Machine for evaluating earthquake-triggered liquefaction. In this research scant dataset of 185 instances had been used to train the model. Also, a prediction algorithm was generated. Before the hybridization of the kernel function, (Kohestani et al., 2015) deployed a random machine forest (Supervised Machine Learning) algorithm for predicting the soil liquefaction potential using CPT case points. He used the uncleaned dataset to develop a prediction model that would not perform satisfactorily on the test data set. In 2015 (Xue & Yang, 2013) utilized the Fuzzy machine learning neural networks for predicting the liquefaction potential of the soil. However, they relied only on Support Vector Machines and the target classes were overlapping, due to which data was not linearly separated.

In 2014 (Muduli & Das) used the Chi–Chi earthquake database to estimate the liquefaction of soil employing genetic programming (GP). The main research gaps left out in their research were: the use of genes and mutation were not identified, and they relied on accuracy parameters only that could be misleading as elucidated in the coming section where accuracy parameters are explained. (Chern et al., 2008) utilized the fuzzy machine learning neural network model for the prediction of soil based on the Cone Penetration Test data points. It has a total of 466 points of data. Based on a fuzzy machine learning neural network model, a fuzzy neural network model known as ANFIS was implemented to predict the soil's potential to liquefy. One main issue faced by the researchers was that a significant amount of noise was prevalently present in the dataset, due to which data remained uncleaned.

In 2008 (Hanna et al.) developed the correlation between soil's liquefaction potential and 12 affecting variables using the deep learning method known as artificial neural networks (ANNs) on 620 cases accumulated from seismic activity cases in Taiwan and Turkey. The developed model has a black-box nature that causes issues regarding the interpretation of the problem at hand. (Pal, 2006) also utilized the CPT and SPT data history records from the already published

literature for predicting the liquefaction of soil. In this research, the developed algorithm showed greater accuracy for the training data and less accuracy for the testing dataset. It is, therefore, not advisable to use algorithms that have high training accuracy and low testing accuracy.

(Rahman & Wang, 2002) also propounded fuzzy neural networks for the prediction of liquefaction potential with the SPT database. The input parameters used for the prediction also contain a cyclic stress ratio. As it has been explained in detail in the previous section, the use of CSR as an input parameter brings a lot of uncertainty that could lead to the uncertain output of the model. This research will not use those input parameters that have inherent uncertainties and generalizations. (In 2003 Baziar & Nilipour,) utilized an Artificial Neural Network including a backpropagation algorithm to ascertain liquefaction potential in different locations based on the results of the Cone Penetration Test. In this research, regression algorithms were used to predict the liquefaction prediction that used a dataset containing the Factor of Safety as an input parameter. As FOS in the liquefaction problem contains certain generalizations, it is not advisable to use a dataset containing FOS.

## Chapter. 3

### 3. METHODOLOGY AND RESEARCH WORK

#### 3.1 Overview of the Machine Learning Algorithms Used

##### 3.1.1 Support Vector Machine

(SVM) is a formidable and flexible learning algorithm that falls under the class of supervised learning, mainly applied to classification tasks and regression tasks. Vapnik and Cortes pioneered SVM in 1995, and it has acquired considerable recognition due to its capacity to handle complex datasets by finding optimal hyperplanes that are suitable for different classes.

#### Principles and Concepts:

- The primary objective of this algorithm is to locate a hyperplane that effectively splits data points of distinct classes. This hyperplane is denoted as  $w \cdot x + b = 0$ ;  $w$  is the weight vector,  $x$  is the input, and  $b$  is the bias.

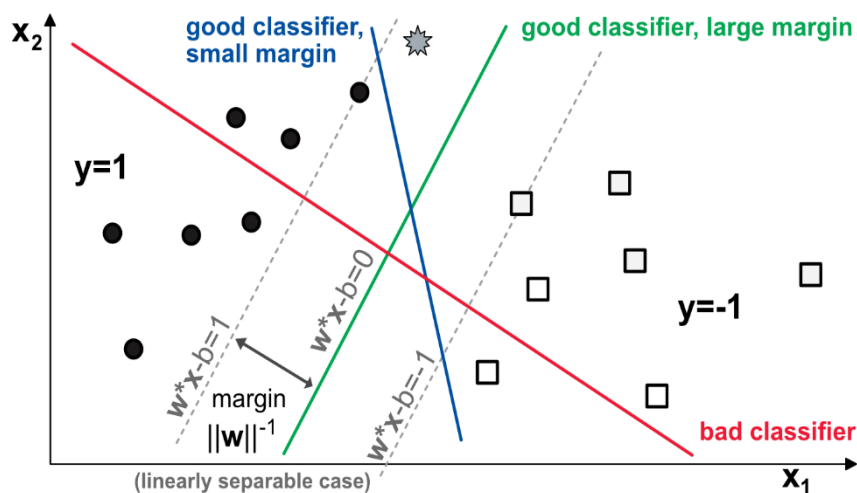


Figure 10 SVM algorithm Graphic Intuition

- **Margin:**

SVM strives to locate the hyperplane with the largest possible boundary connecting the two groups. The maximum distance between the hyperplane and the closest data points

from each class is equal to the maximum. Maximizing this margin enhances the generalization capabilities of the model and reduces the risk of overfitting.

- **Support Vectors:**

The data points that lie on the margins or within the margin boundary are called support vectors. These vectors are crucial as they influence the placement of the optimal hyperplane. SVM focuses on these points to make accurate predictions.

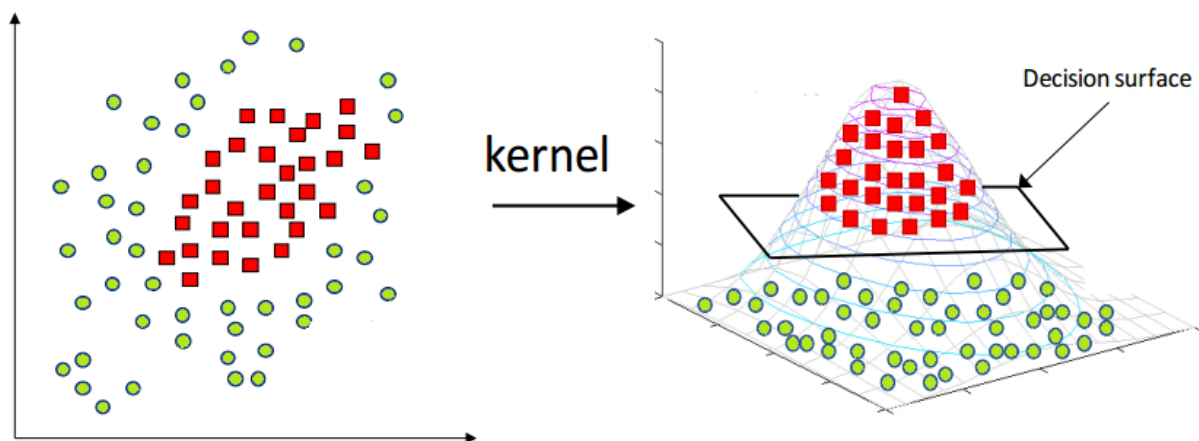
**Working Principle:**

- **Linear SVM:**

In a linearly separable scenario, SVM aims to find the hyperplane with the maximum margin.

- **Non-Linear SVM:**

When the data is not linearly separable, SVM utilizes kernel functions to transform the data into a higher-dimensional space. In this transformed space, a linear decision boundary can potentially separate the classes.



*Figure 11 Kernel Function for non-linearity*

**Advantages of SVM:**

- **Robust to High Dimensional Data:**

SVM is effective in high-dimensional spaces, which is especially useful for tasks such as text categorization, classification, and image recognition.

- **Handles non-linearity:**

The kernel trick allows SVM to model complex relationships that cannot be captured by linear models.

- **Control on Regularization:**

The 'C' parameter provides control over the trade-off between margin maximization and classification error minimization, thus preventing overfitting.

- **Global Optimal Solution:**

SVM optimization aims for a global optimal solution, which contributes to its stability and reliability.

### *3.1.2 Logistic Regression*

#### **Introduction:**

Logistic Regression is a widely used method for binary classification, particularly suited for situations where the dependent variable is categorical, and the goal is to predict the probability of an event occurring. It is a fundamental algorithm in machine learning and serves as a building block for more complex models.

#### **Key Concepts:**

- **Sigmoid Function:**

The core idea behind logistic regression is the use of the sigmoid (logistic) function to represent the linear sequence of input characteristics to a value between 0 and 1. This value corresponds to the anticipated likelihood that the dependent variable belongs to the positive class.

$$p(y = 1/x) = \frac{1}{1 + e^{-z}} \quad (4)$$

**Formulation:**

The logistic regression model is formulated as follows:

$$\log(odds) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \dots + \beta_k * x_k \quad (5)$$

Where,

**log(odds)** is the log odds of the positive class.

$x_1, x_2, \dots, x_k$  are the input features.

$\beta_0, \beta_1, \beta_2, \dots, \beta_k$  are the coefficients corresponding to each feature

- **Objective Function (Likelihood):**

The goal of logistic regression is to estimate the coefficients that maximize the likelihood of observing the given data under the model. The likelihood function represents the probability of observing the given set of outcomes (dependent variable) given the predictor variables and a set of model parameters. For a binary logistic regression model, where the dependent variable  $y$  takes on values of 0 or 1, the likelihood function for a single observation

$$L(\beta) = \pi_i (p(y = 1|x_i))^{y_i} * (1 - p(y = 1|x_i))^{(1-y_i)} \quad (6)$$

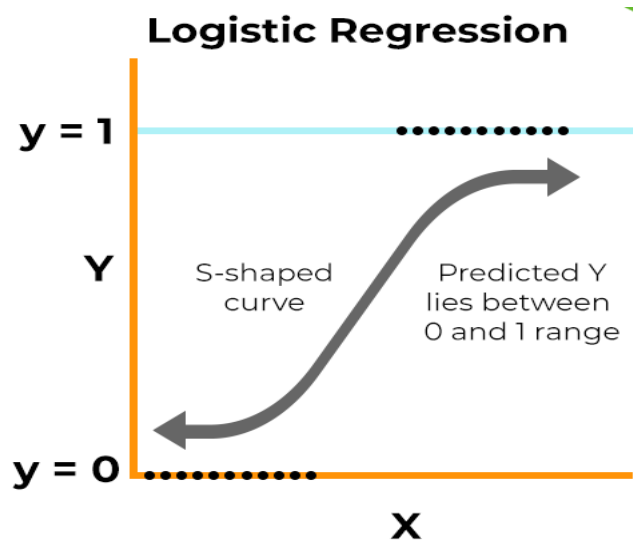


Figure 12 Logistic Regression for Binary Classification

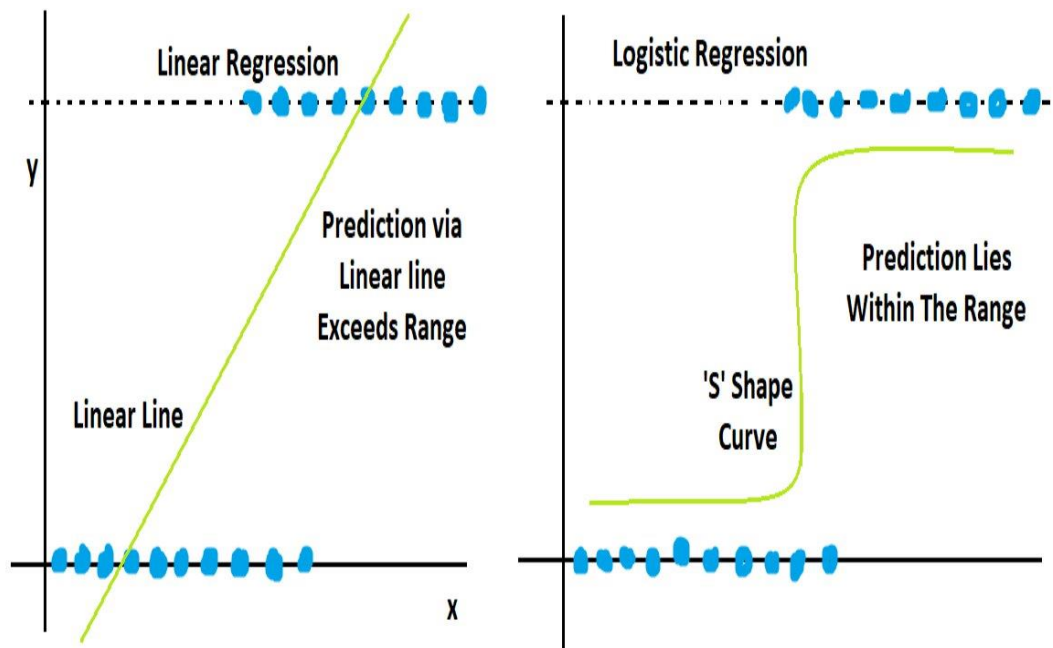


Figure 13 Graphical Intuition of Logistic Regression

**Advantages:**

- **Simple Interpretability:**

Logistic regression coefficients can be interpreted as the change in log odds for a unit change in the corresponding feature.

- **Efficiency:**

Logistic regression is computationally efficient and works well for linearly separable data

**Limitations:**

- **Limited Complexity:**

Logistic regression may not perform well on complex datasets with intricate decision boundaries.

- **Sensitive to Outliers:**

Outliers can significantly impact the coefficients and predictions.



### 3.1.3 Decision Trees

#### **Introduction:**

Decision Trees are a versatile and widely used machine algorithm used for both classification tasks and regression tasks. They represent a graphical model of decisions and their possible consequences in a tree-like structure. Decision Trees are intuitive, and interpretable, and can capture complex relationships in data, making them a valuable tool in data analysis and predictive modeling.

#### **Key Concepts:**

- **Nodes and Edges:**

A decision tree contains nodes & edges. Nodes denote decisions or test conditions, while edges connect nodes and indicate the outcomes of the tests.

- **Root Node:**

The highest node of the tree represents the initial decision point where the first test is applied.

- **Internal Nodes:**

Every node other than the root node is an internal node, representing intermediate decisions.

- **Leaf Nodes:**

Terminal nodes, or leaf nodes, represent final decisions or outcomes, such as class labels in a classification task or predicted values in a regression task.

- **Splitting Criteria:**

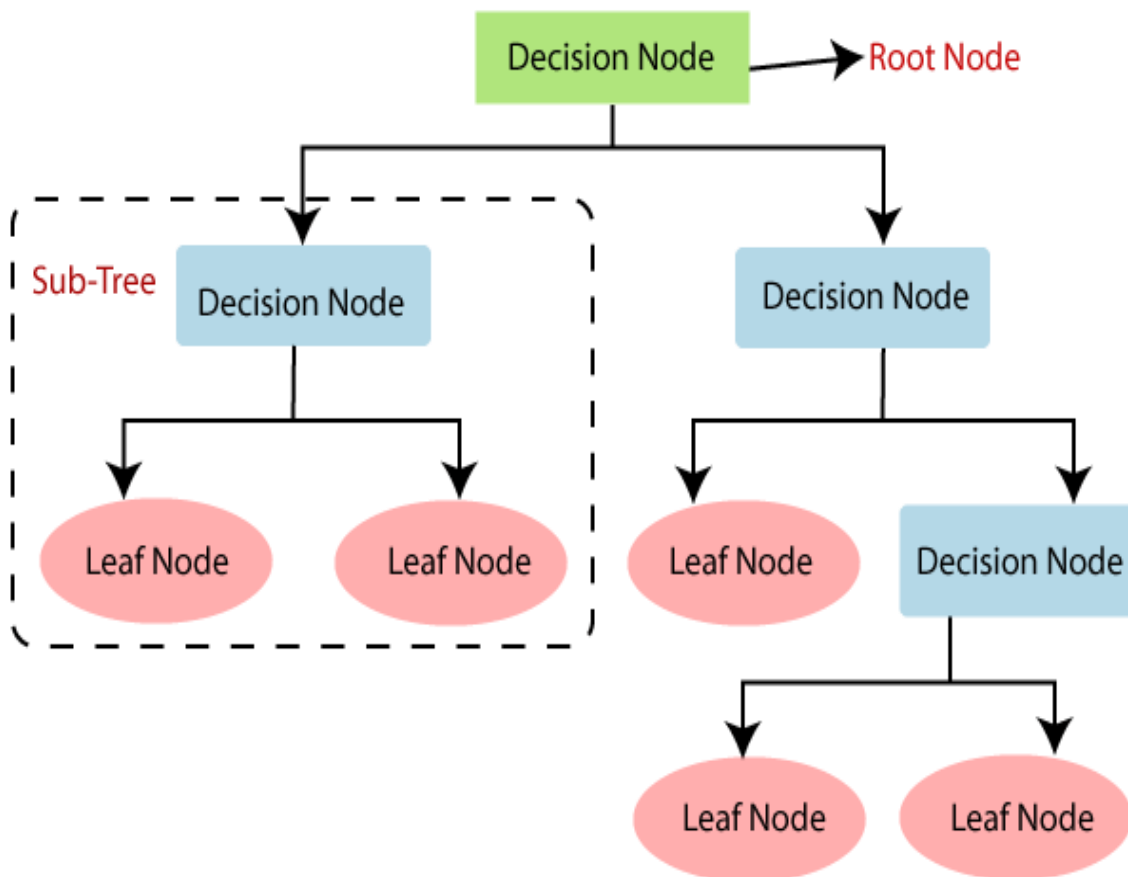
The process of creating decision nodes involves selecting features and values that optimize a splitting criterion, typically aimed at reducing impurity in classification tasks or minimizing variance in regression tasks.

**Formulation:**

For a node 't' containing data points of class 'i' with proportion 'p(i)', the Gini impurity is calculated as:

$$\text{Gini}(t) = 1 - \sum [p(i|t)]^2 \tag{7}$$

For a node 't' containing data points of class 'i' with proportion 'p(i)', the entropy is calculated



*Figure 14 Decision Tree Model Intuition*

**Advantages:**

- **Interpretability:**

DTs are easily interpreted, making them suitable for presenting decisions to stakeholders.

- **Handling Irrelevant Features:**

Decision Trees can automatically select important features and disregard irrelevant ones.

### **Limitations:**

- **Overfitting:**

Deep trees can overfit noisy data, leading to poor generalization of new data.

- **Predisposition towards Dominant Classes:**

DTs tend to favor dominant classes, potentially causing imbalanced data problems. (Jas & Dodagoudar, 2023)

### *3.1.4 Artificial Neural Network:*

#### **Introduction:**

(ANNs) are a class of learning models inspired by the biological neurons in the human brain. ANNs consist of joined nodes, or artificial neurons, ordered in layers. They excel at learning patterns and relationships in data, enabling them to perform tasks.

#### **Key Concepts:**

- **Neuron Model:**

An artificial neuron processes input data and produces an output using weights and biases. The output is determined by a function applied to the sum of inputs and biases.

(Jas & Dodagoudar, 2023)

Mathematically, the output 'a' of a neuron with 'n' inputs is:

$$a = \text{activation}(\sum(\text{weight } i * \text{input } i) + \text{bias}) \quad (8)$$

- **Activation Functions:**

Activation functions initiate non-linearity into the network, allowing it to model intricate connections.

Common activation functions include:

$$\text{sigmoid: } \frac{1}{(1 + \exp(-x))} \quad (9)$$

$$\text{hyperbolic tangent (tanh)} = \frac{(\exp(-x) - \exp(x))}{(\exp(x) + \exp(-x))} \quad (10)$$

### Types of Layers:

- **Input Layer:**

Receives input data, features, or observations. It serves as the interface between the external environment or dataset and the network itself. The number of nodes in the input layer is determined by the dimensionality of the input data.

- **Hidden Layers:**

Intermediate layers between the input and output layers. They learn representations and transform data. In neural networks, hidden layers are layers of nodes between the input and output layers where computation occurs. The types of hidden layers in neural networks can vary based on their architecture and function

- **Output Layer:**

Produces the final prediction or output of the network. It represents the results or predictions generated by the network based on the input data and the learned parameters. The structure and characteristics of the output layer depend on the nature of the task the neural network is designed to solve.

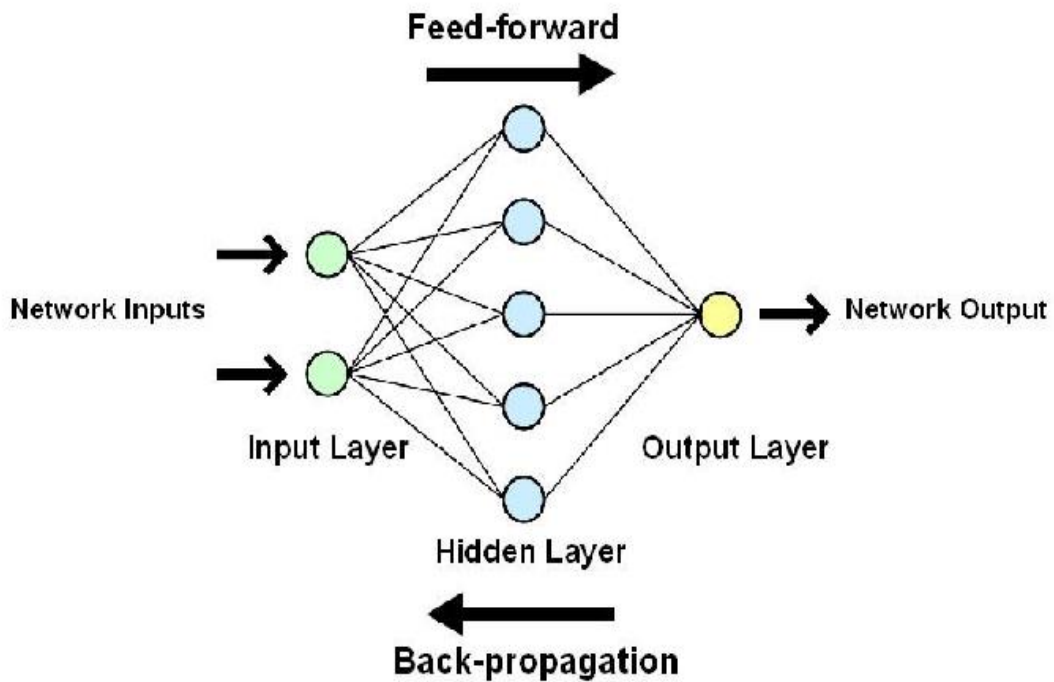


Figure 15 Artificial Neural Network Structure

**Advantages:**

- **Pattern Learning:**

ANNs can learn intricate patterns from data, making them suitable for complex tasks

- **Feature Extraction:**

They can automatically learn relevant features from raw data, reducing the need for manual feature engineering

- **Versatility:**

ANNs can handle a wide range of data types and solve various tasks.

**Limitations:**

- **Data Requirements:**

ANNs require substantial amounts of data for effective training

- **Overfitting:**

Deep networks overfit small datasets, requiring regularization techniques.

## 4. DATASET DESCRIPTION AND EXPLORATORY DATA ANALYSIS

### 4.1 Dataset Description:

The dataset utilized in this study has been gathered and collected from previous research papers. This dataset has a total of 226 observations of liquefaction and non-liquefaction instances.

This dataset has been gathered from the following research papers. (Juang et al., 2003) (Goh & Goh, 2007) (Baziar & Nilipour, 2003). A total of six input variables have been used in this study to develop machine learning algorithms. These input variables are:

- Cone tip resistance
- Sleeve friction ratio
- Effective Stress
- Total Stress
- Maximum horizontal ground surface acceleration
- Earthquake moment magnitude

```
df.head()
```

|   | qc_Mpa | R f % | eff_stress | total_stress | a max | Mw  | Liquefaction |
|---|--------|-------|------------|--------------|-------|-----|--------------|
| 0 | 2.6    | 3.3   | 103.5      | 116.0        | 0.50  | 6.4 | 1            |
| 1 | 12.9   | 3.5   | 215.2      | 267.2        | 0.50  | 6.4 | 0            |
| 2 | 6.8    | 4.9   | 173.5      | 274.0        | 0.50  | 6.4 | 0            |
| 3 | 7.5    | 4.5   | 171.0      | 254.5        | 0.50  | 6.4 | 0            |
| 4 | 5.1    | 0.4   | 66.2       | 121.2        | 0.13  | 7.3 | 1            |

*Figure 16 Data head in Jupyter Notebook*

## 4.2 Statistical Description of Data

Statistical description of data involves summarizing and analyzing various aspects of a dataset to gain insights and comprehend its characteristics. Jupyter Notebook has been used to determine the statistical parameters of the data. It has been done by using the function (describe) in the jupyter notebook.

```
df.describe()
```

|       | qc_Mpa     | R f %      | e-stress   | total-stress | a max      | Mw         | Liquefaction |
|-------|------------|------------|------------|--------------|------------|------------|--------------|
| count | 226.000000 | 226.000000 | 226.000000 | 226.000000   | 226.000000 | 226.000000 | 226.000000   |
| mean  | 5.817257   | 1.217699   | 74.648673  | 106.890708   | 0.289292   | 6.946018   | 0.588496     |
| std   | 4.092281   | 1.048023   | 34.395425  | 55.359292    | 0.144060   | 0.438388   | 0.493199     |
| min   | 0.900000   | 0.100000   | 22.500000  | 26.600000    | 0.080000   | 6.000000   | 0.000000     |
| 25%   | 3.000000   | 0.500000   | 51.800000  | 67.725000    | 0.190000   | 6.600000   | 0.000000     |
| 50%   | 4.900000   | 0.900000   | 62.800000  | 90.300000    | 0.250000   | 7.100000   | 1.000000     |
| 75%   | 7.500000   | 1.775000   | 96.775000  | 128.600000   | 0.370000   | 7.100000   | 1.000000     |
| max   | 25.000000  | 5.200000   | 215.200000 | 274.000000   | 0.800000   | 7.600000   | 1.000000     |

*Figure 17 Statistical Description of Input Data*

### 4.2.1 Correlation Matrix

A correlation matrix is a valuable statistical tool used to comprehend the relationships between multiple variables in a dataset. It provides insights into how pairs of variables move together, which can be crucial for making informed decisions.

In a correlation matrix, each cell contains the correlation coefficient between two variables.

Understanding the values in a correlation matrix:

- A positive correlation coefficient suggests that as one variable increases, the other tends to increase as well.
- A negative correlation coefficient indicates that as one variable increases, the other tends to decrease.
- A correlation coefficient close to 0 suggests little to no linear relationship between the variables.

A well-constructed correlation matrix facilitates researchers to identify patterns and potential multicollinearity (high correlation between predictor variables) in datasets, which can impact statistical analyses.

```
df.corr()
```

|              | qc_Mpa    | R f %     | eff-stress | total-stress | a max     | Mw        | Liquefaction |
|--------------|-----------|-----------|------------|--------------|-----------|-----------|--------------|
| qc_Mpa       | 1.000000  | -0.271072 | 0.250165   | 0.248720     | 0.049122  | -0.085369 | -0.458918    |
| R f %        | -0.271072 | 1.000000  | 0.315974   | 0.371510     | 0.031699  | -0.105288 | -0.278197    |
| eff-stress   | 0.250165  | 0.315974  | 1.000000   | 0.925128     | 0.273245  | 0.038879  | -0.185775    |
| total-stress | 0.248720  | 0.371510  | 0.925128   | 1.000000     | 0.110358  | 0.033804  | -0.258394    |
| a max        | 0.049122  | 0.031699  | 0.273245   | 0.110358     | 1.000000  | -0.101525 | 0.374331     |
| Mw           | -0.085369 | -0.105288 | 0.038879   | 0.033804     | -0.101525 | 1.000000  | 0.092083     |
| Liquefaction | -0.458918 | -0.278197 | -0.185775  | -0.258394    | 0.374331  | 0.092083  | 1.000000     |

Figure 18 Correlation Matrix of the Data set



```
sns.heatmap(df.corr(), cmap='coolwarm', annot=True)
plt.title('Correlation Matrix')
```

```
Text(0.5, 1.0, 'Correlation Matrix')
```

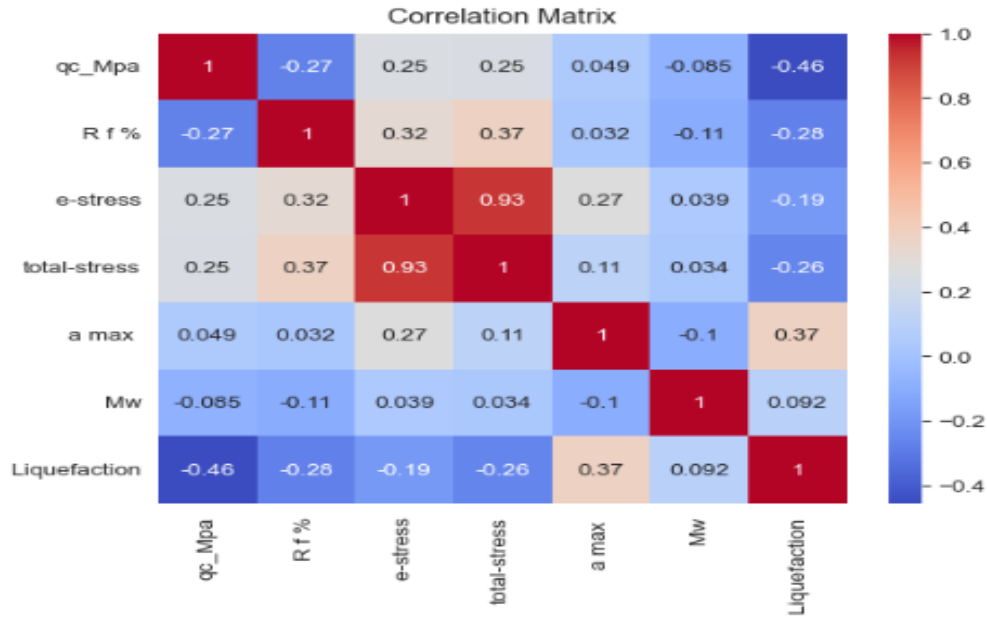


Figure 19 Heatmap of the Correlation Matrix

```
df.corr()['Liquefaction'].sort_values().plot(kind='bar')
plt.title('Correlation by Bar Graph')
```

```
Text(0.5, 1.0, 'Correlation by Bar Graph')
```

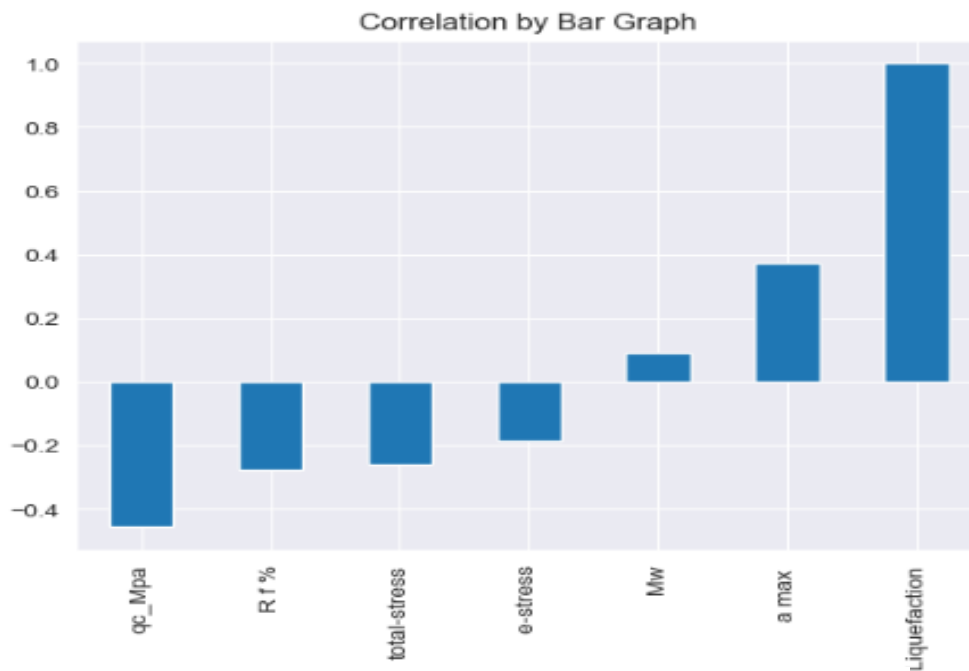


Figure 20 Bar Graph of the Correlation Matrix

## 4.3 Exploratory Data Analysis

(EDA) is an indispensable and effective technique in data analysis that is premised upon understanding the structure, patterns, and potential information within a dataset. EDA involves a range of techniques to unearth hidden information, detect anomalies, and formulate hypotheses before more formal statistical analysis or modeling begins.

### 4.3.1 Histogram

A histogram is a visual portrayal of the distribution of a dataset. It gives a visual overview of the frequency of values within the specified intervals, called "bins".

```
sns.histplot(df['qc_Mpa'],bins=30,kde=True)  
plt.title('Tip Resistance Histogram')
```

```
Text(0.5, 1.0, 'Tip Resistance Histogram')
```

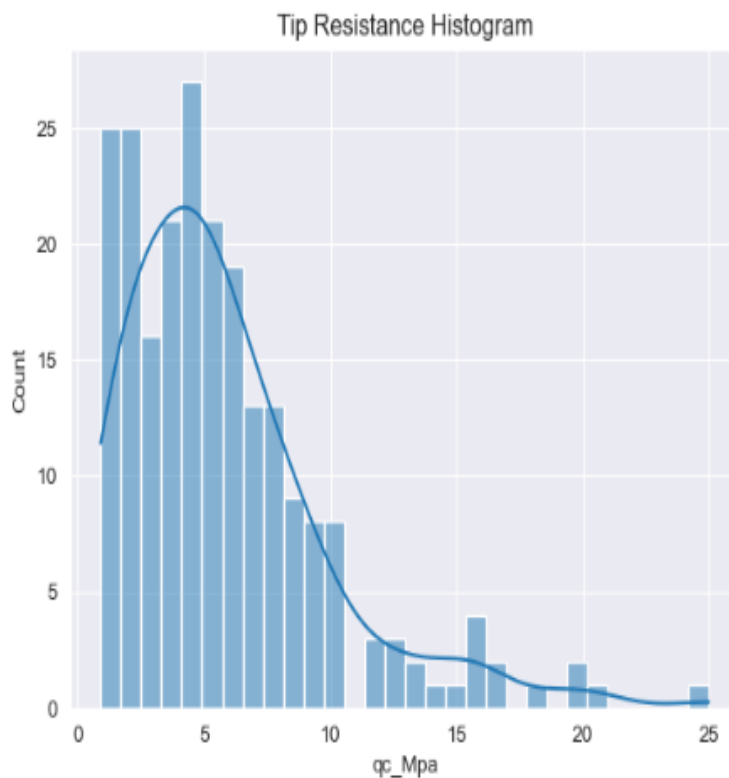


Figure 21 Tip Resistance Histogram

```
sns.histplot(df['R f %'],bins=30,kde=True)
plt.title('Sleeve Friction Histogram')
```

```
Text(0.5, 1.0, 'Sleeve Friction Histogram')
```

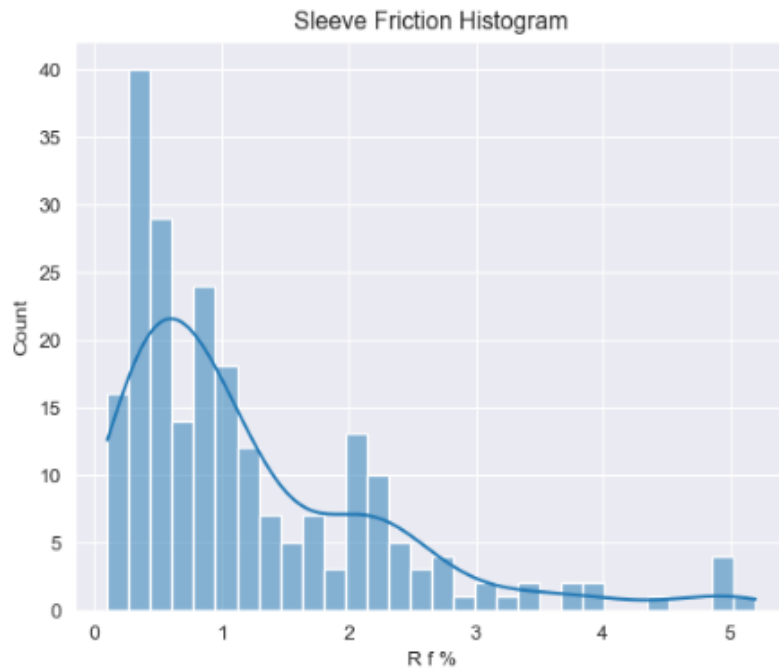


Figure 22 Sleeve Friction Histogram

```
sns.histplot(df[' e-stress'],bins=30,kde=True)
plt.title('Effective Stress Histogram')
```

```
Text(0.5, 1.0, 'Effective Stress Histogram')
```

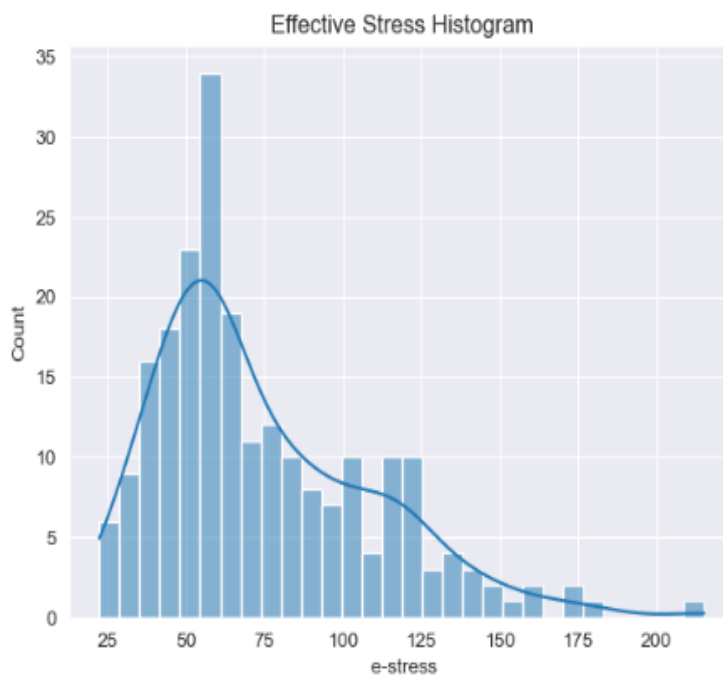


Figure 23 Effective Stress Histogram

### 4.3.2 Joint plots

A joint plot is a dynamic data visualization tool in Python, which is often created using libraries such as Seaborn or Matplotlib. It combines multiple univariate and bivariate plots to provide an overview of the relationship between two variables. A joint plot typically includes scatter plots and a kernel density estimate.

```
sns.jointplot(x='qc_Mpa',y='total-stress',data=df,kind='scatter',hue='Liquefaction')  
<seaborn.axisgrid.JointGrid at 0x225d9d0b150>
```

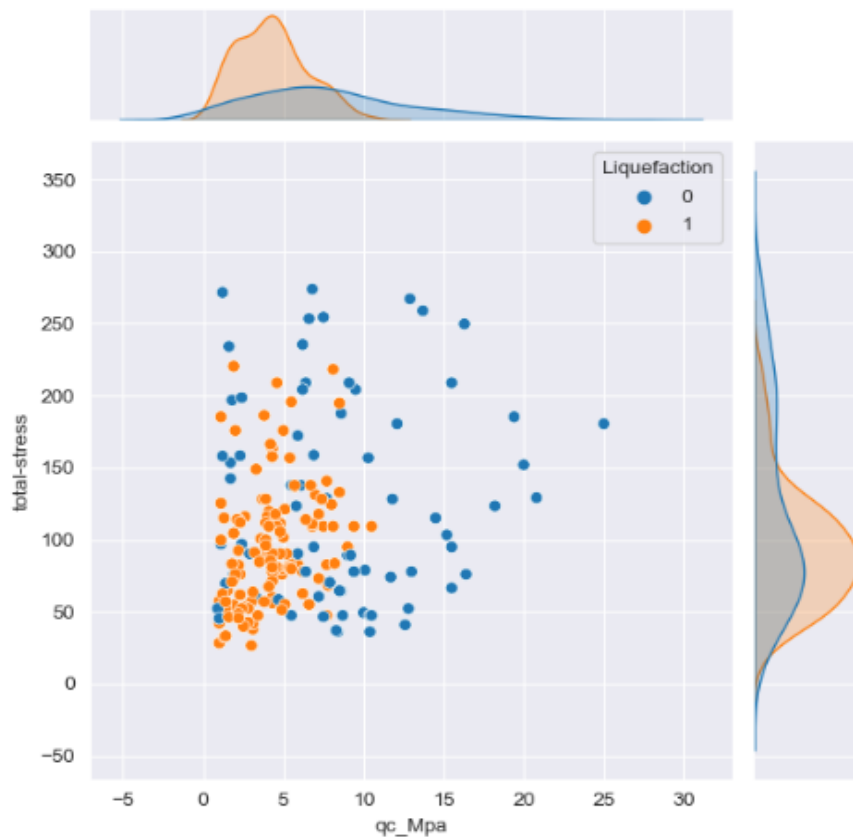


Figure 24 Joint plot of Total stress and Cone Resistance

It can be observed that after a certain value of Cone Penetration is achieved, that is 12 Mpa, the liquefaction potential of the soil diametrically decreases. This plot further corroborates the finding that with the increase in Cone Penetration Resistance, liquefaction potential decreases.

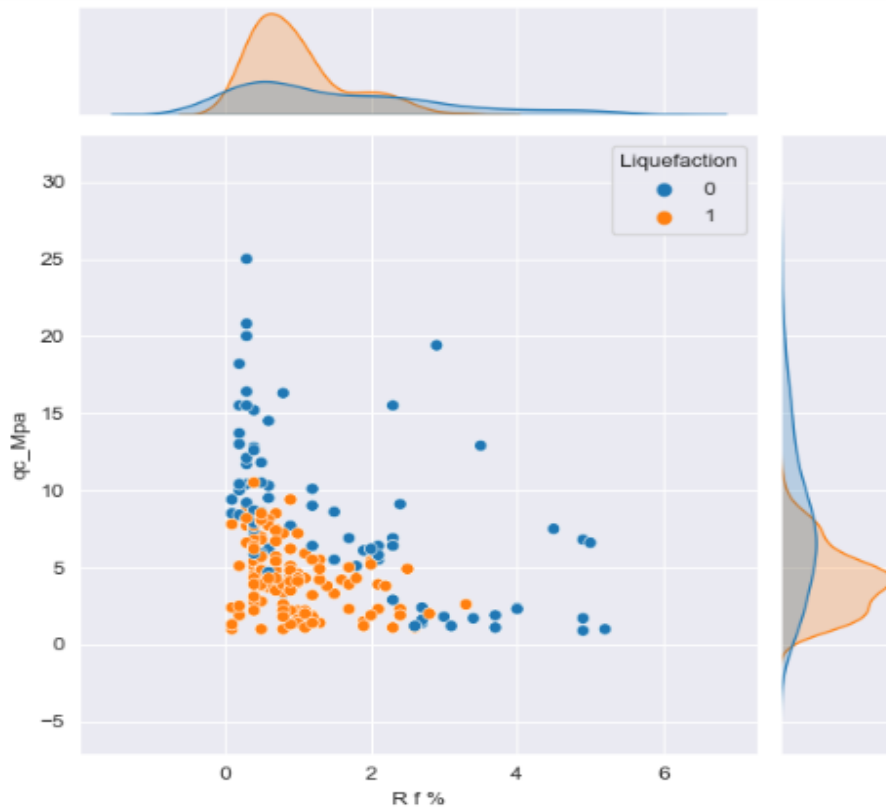


Figure 25 Joint plot of Sleeve Friction and Cone Resistance

### 4.3.3 Box Plots

A box plot is a graphical demonstration of the allocation of a dataset by five key summary statistics: the minimum, first quartile (Q1), median, third quartile (Q3), and maximum.

```
sns.boxplot(data=df)
plt.title('Box Plot of Input Features')
Text(0.5, 1.0, 'Box Plot of Input Features')
```

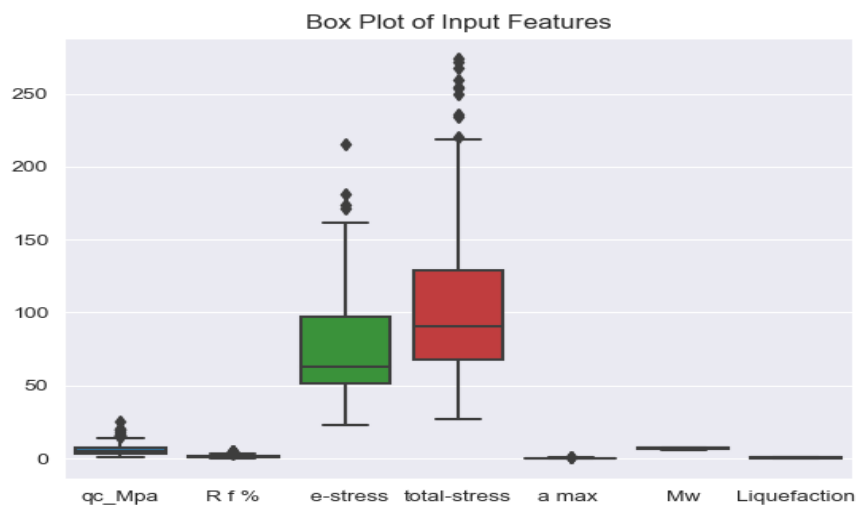
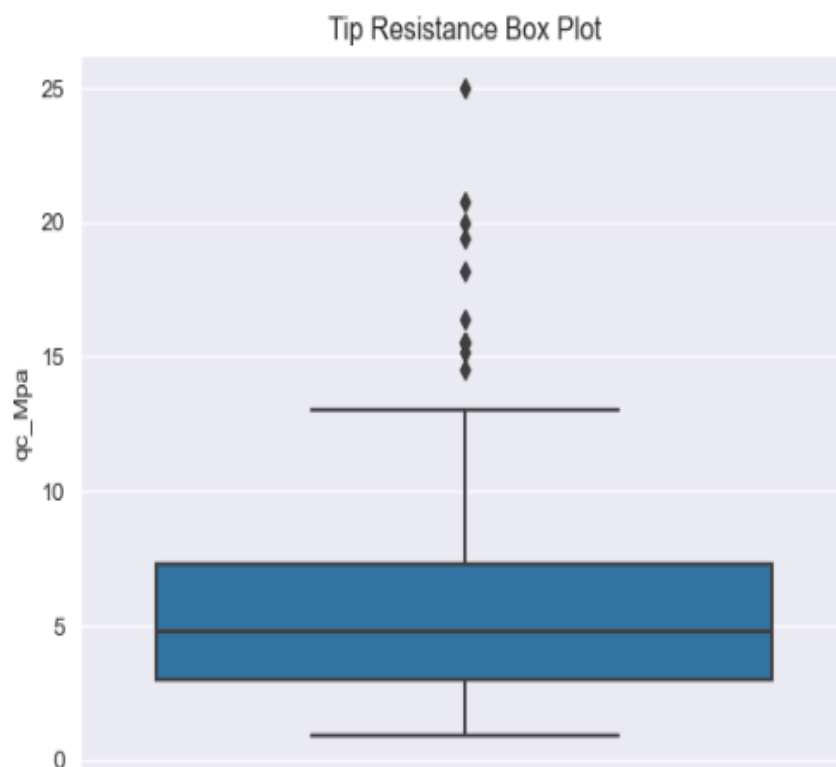


Figure 26 Boxplot of Input Variables

As can be seen from the above box plot, there are considerable numbers of outliers present in total stress and cone penetration resistance. The presence of these outliers can, sometimes, alter the desired results by shifting the central tendency. It is, therefore, indispensable to remove the outliers present in the data so that the data can be cleaned, and machine learning algorithms can give a good performance.

```
sns.boxplot(y='qc_Mpa',data=df1)  
plt.title('Tip Resistance Box Plot')
```

```
Text(0.5, 1.0, 'Tip Resistance Box Plot')
```



*Figure 27 Tip Resistance Box Plot*

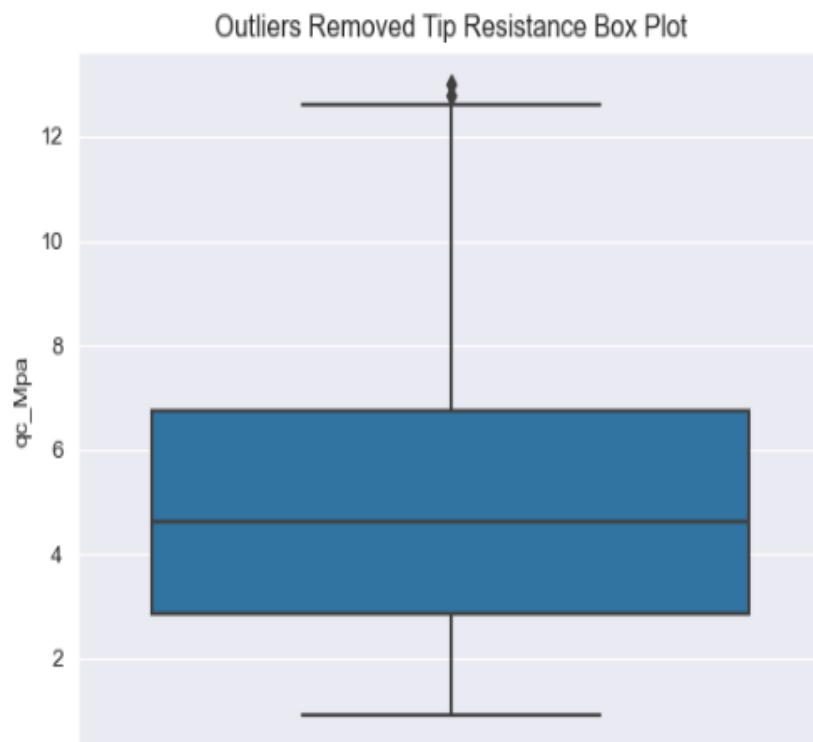
It can be seen from above Figure 31 that the tip resistance data contains a considerable number of outliers. Removal of these outliers can give a better result while performing machine learning algorithms. A code, shown below, was generated to remove the outliers from the Tip resistance dataset.

```
df['qc_Mpa'].mean()  
dat2=dat1[dat1['qc_Mpa']<13]
```

```
sns.boxplot(data=dat2)  
plt.title('Outliers Removed Box Plot')
```

```
sns.boxplot(y='qc_Mpa', data=df2)  
plt.title('Outliers Removed Tip Resistance Box Plot')
```

```
Text(0.5, 1.0, 'Outliers Removed Tip Resistance Box Plot')
```

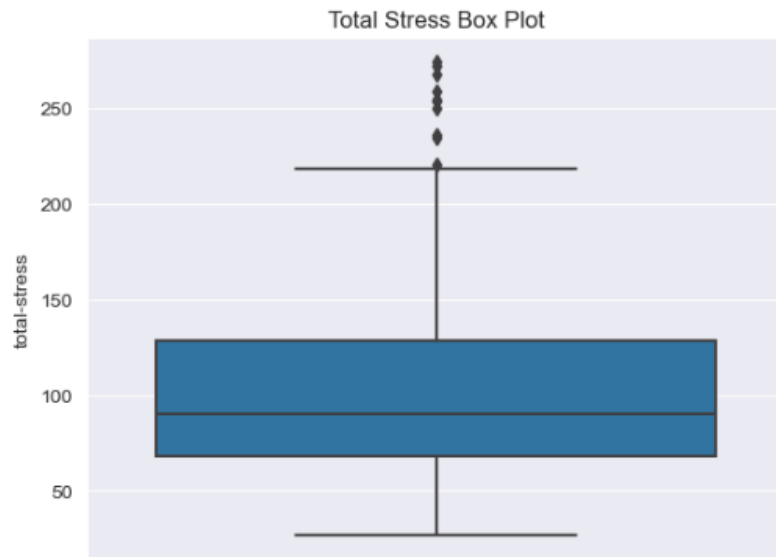


*Figure 28 Outliers Removed Tip Resistance Box*

Similarly, outliers were removed from the total stress data so that the data becomes clean, and the performance of machine learning algorithms can be optimized. The process of removing the outliers has been a monumental activity to train the particular set of algorithms, as outliers can significantly alter the performance of the algorithm.

```
sns.boxplot(y='total-stress',data=df)
plt.title('Total Stress Box Plot')
```

```
Text(0.5, 1.0, 'Total Stress Box Plot')
```



*Figure 29 Total Stress Box Plot*

```
sns.boxplot(y='total-stress',data=df1)
plt.title('Outliers removed Total Stress Box Plot')
```

```
Text(0.5, 1.0, 'Outliers removed Total Stress Box Plot')
```



*Figure 30 Outliers Removed Total Stress Box Plot*



#### 4.3.4 Swarm Plot

A swarm plot is a method of data visualization that provides a unique way to display the distribution of categorical data along with individual data points. In contrast to traditional scatter plots, a swarm plot is specifically designed for categorical variables and aims to prevent data points from overlapping. This technique results in a clearer representation of data density and distribution within each category. In a swarm plot, each data point is arranged individually along the categorical axis. The positioning of data points is adjusted to avoid overlapping, creating a visual pattern that appears to be a "swarm" of points.

```
sns.swarmplot(x="Liquefaction", y='qc_Mpa', data=df, hue='Liquefaction')  
plt.title('Swarm Plot of Liquefaction vs Tip Resistance')
```

```
Text(0.5, 1.0, 'Swarm Plot of Liquefaction vs Tip Resistance')
```

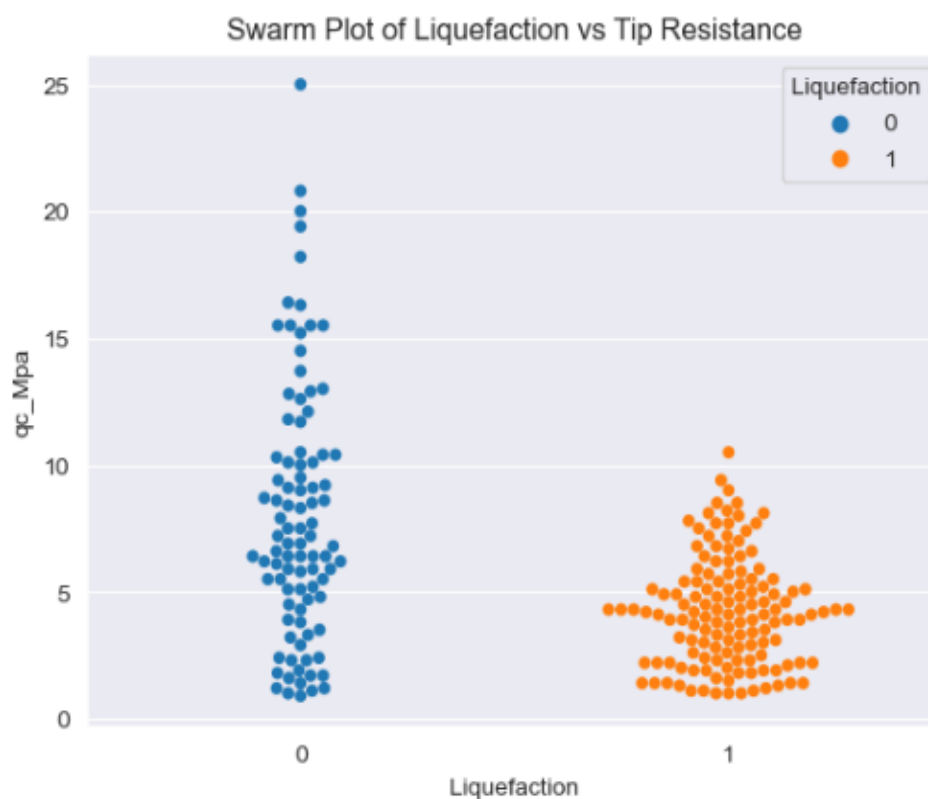


Figure 31 Swarm Plot of the data points

### 4.3.5 Pair plot

A joint plot is a dynamic data visualization in Python, created using libraries such as Seaborn or Matplotlib. It combines multiple plots to provide a comprehensive view of the relationship among variables. A joint plot typically includes scatter plots, histograms, and sometimes a regression line or a kernel density estimate.

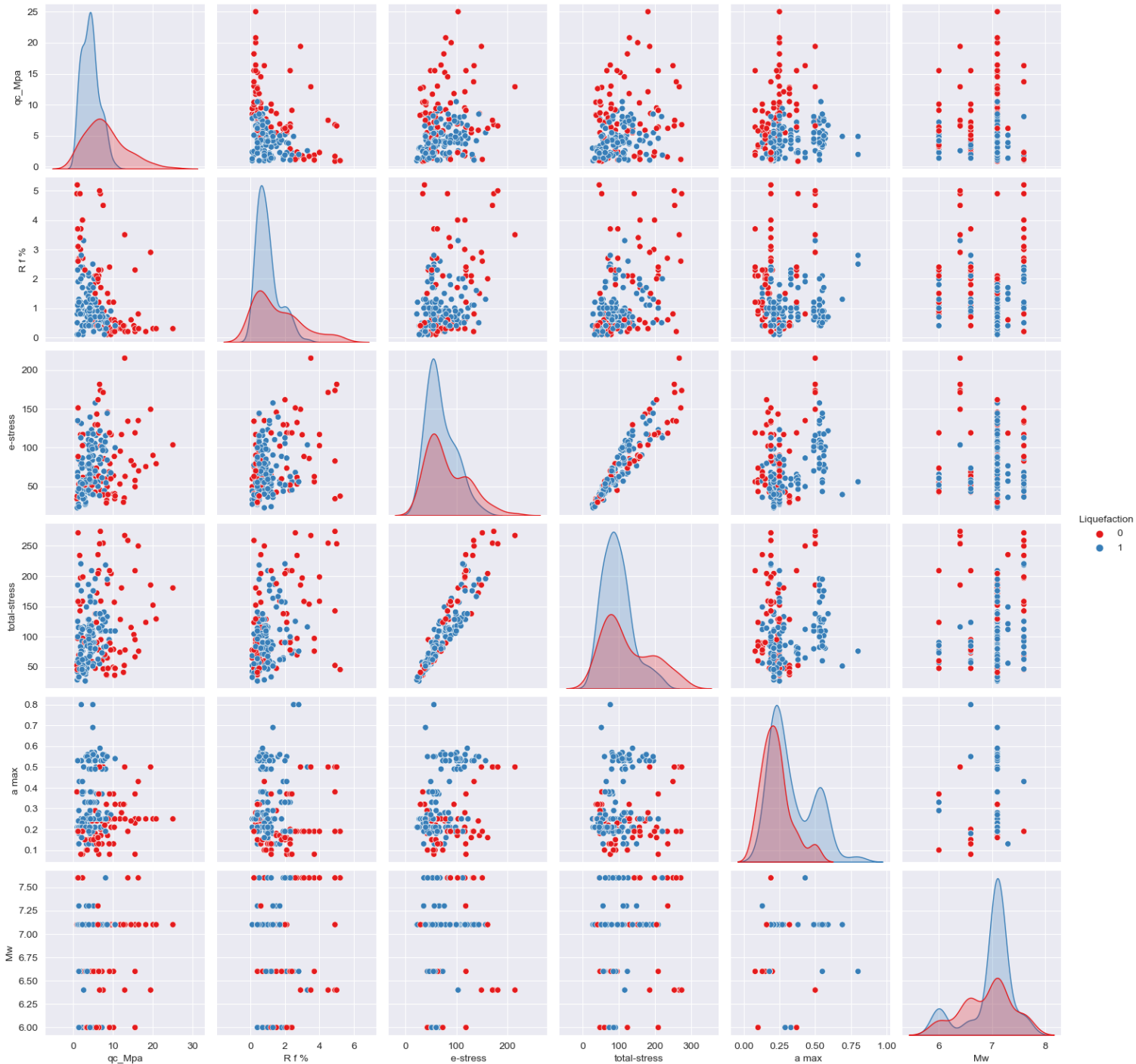


Figure 32 Pair plot of the datasets

## **Chapter. 5**

### **5. PERFORMANCE EVALUATING PARAMETERS**

The performance evaluation of classification algorithms involves assessing how well a model's predictions match the actual results. Various metrics and parameters are used to determine the quality and effectiveness of these algorithms. Here are key performance-determining parameters for classification algorithms.

#### **5.1 Confusion Matrix**

A confusion matrix is a table that lists the outcomes of a classification problem. It illustrates a comparison between the predicted classes generated by a model and the actual classes present in the dataset. The matrix is typically arranged as a 2x2 table for binary classification tasks, with each cell representing different combinations of true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

##### **Key Elements Interpretation:**

**i. True Positive (TP):**

Those datasets that are correctly predicted as positive class are known as True positive class.

**ii. True Negative (TN):**

Those datasets that are correctly predicted as negative class are known as True negative class.

**iii. False Positive (FP):**

Instances that are predicted as a positive class but belong to the negative class. This is also known as Type I error.

**iv. False Negative (FN):**

Instances that are predicted as a negative class but belong to the positive class. This is also known as Type II error.

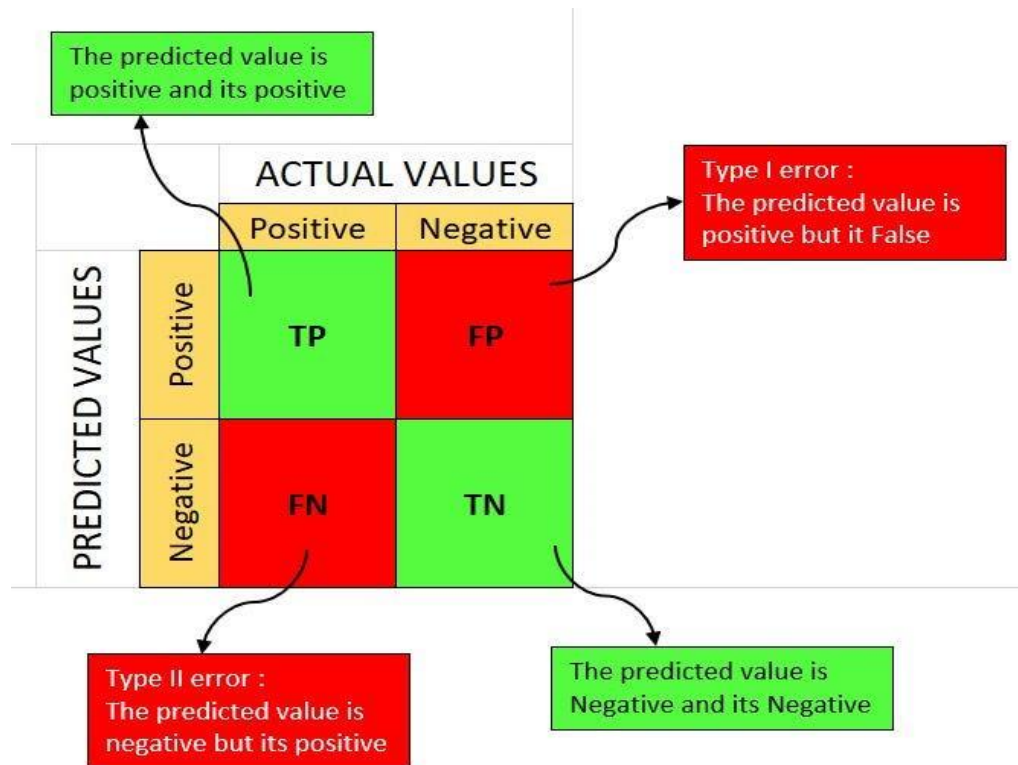


Figure 33 Confusion Matrix Binary Classification

## 5.2 Accuracy

The proportion of correctly guessed cases to the total number of cases in the dataset. While it provides an overall idea of the model's performance, accuracy might not be appropriate for imbalanced datasets. *The accuracy score does not specifically tell you about the mistake being made by the model. Hence, it does not tell you which outcome is incorrectly predicted.*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

### 5.3 Precision

Precision is the fraction of true positive estimates to the sum of true positive predictions and false positive predictions. It quantifies the accuracy of the positive predictions made by the model, specifically focusing on how well it avoids making incorrect positive predictions. It indicates the accuracy of positive predictions. It tells you what prediction of predicted positive is truly positive.

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

### 5.4 Recall

It is defined as True Positive divided by the sum of True Positive and False Negative. It indicates the model's ability to capture positive instances.

$$Recall = \frac{TP}{TP + FN} \quad (13)$$

### 5.5 F1 Score

The F1 score is the harmonic mean of precision and recall. It considers both false positives and false negatives, offering a balanced assessment of a classification model's effectiveness. The F1 score is especially useful when the class distribution is skewed or when the consequences of false positives and false negatives differ significantly. It is used when it cannot be decided which class is more dangerous than the other class.

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (14)$$

## ***Chapter. 6***

### **6. PYTHON PROGRAMMING, RESULTS AND DISCUSSIONS**

#### **6.1 Python Programming**

In this research work Anaconda environment has been used commensurate with Jupyter Notebook to write the Python codes. Four different machine learning algorithms have been used to predict the liquefaction potential. The details of these algorithms have already been discussed in the previous sections. In this section, different programs that have been made will be displayed and the subsequent performance of the algorithms will be discussed.

##### *6.1.1 Logistic Regression Classification*

The following steps have been taken to perform the Logistic Regression Classification.

- a. Importing different libraries for data extraction, data visualization, and machine learning algorithms
- b. These libraries are:
  - i. Pandas Library
  - ii. NumPy Library
  - iii. Seaborn Library
  - iv. Matplotlib.pyplot Library
  - v. TensorFlow Library

After importing the aforementioned libraries in the Jupyter Notebook, programs were made to analyze, visualize, and predict the data using the Logistic Regression Algorithm.

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

```
In [2]: df=pd.read_csv('Tip resistance data.csv')
```

After importing the necessary libraries in the Jupyter Notebook, sklearn. the model library was used to train, test, and split the data in the Python notebook. I have used a test size of 0.30, which means 30 percent of the data has been used to test the model's performance and 70 percent has been used to train the particular algorithm.

```
In [16]: from sklearn.model_selection import train_test_split
```

```
In [17]: X_train, X_test, y_train, y_test = train_test_split(df1.drop('Liquefaction',axis=1),
                                                            df1['Liquefaction'], test_size=0.30,
                                                            random_state=2)
```

```
In [18]: from sklearn.linear_model import LogisticRegression
logmodel = LogisticRegression()
logmodel.fit(X_train,y_train)
```

```
Out[18]: ▾ LogisticRegression
LogisticRegression()
```

```
In [19]: predictions = logmodel.predict(X_test)
```

```
In [20]: from sklearn.metrics import classification_report,confusion_matrix
cm=confusion_matrix(y_test,predictions)
print(cm)
print(classification_report(y_test,predictions))
```

...

```
In [21]: print(confusion_matrix(y_test,predictions))
```

...

```
In [22]: sns.heatmap(cm ,cmap='Greens',annot=True)
plt.title('Confusion Matrix')
```

*Figure 34 Python Code of Logistic Regression*

### 6.1.2 Logistic Regression Results

As it has already been discussed in detail the performance of binary classification in machine learning is evaluated based on certain parameters. These parameters are, and are not limited to, Accuracy, Precision, Recall, and F1 Score. All these parameters are evaluated based on a confusion matrix which is also discussed in detail in the previous sections.

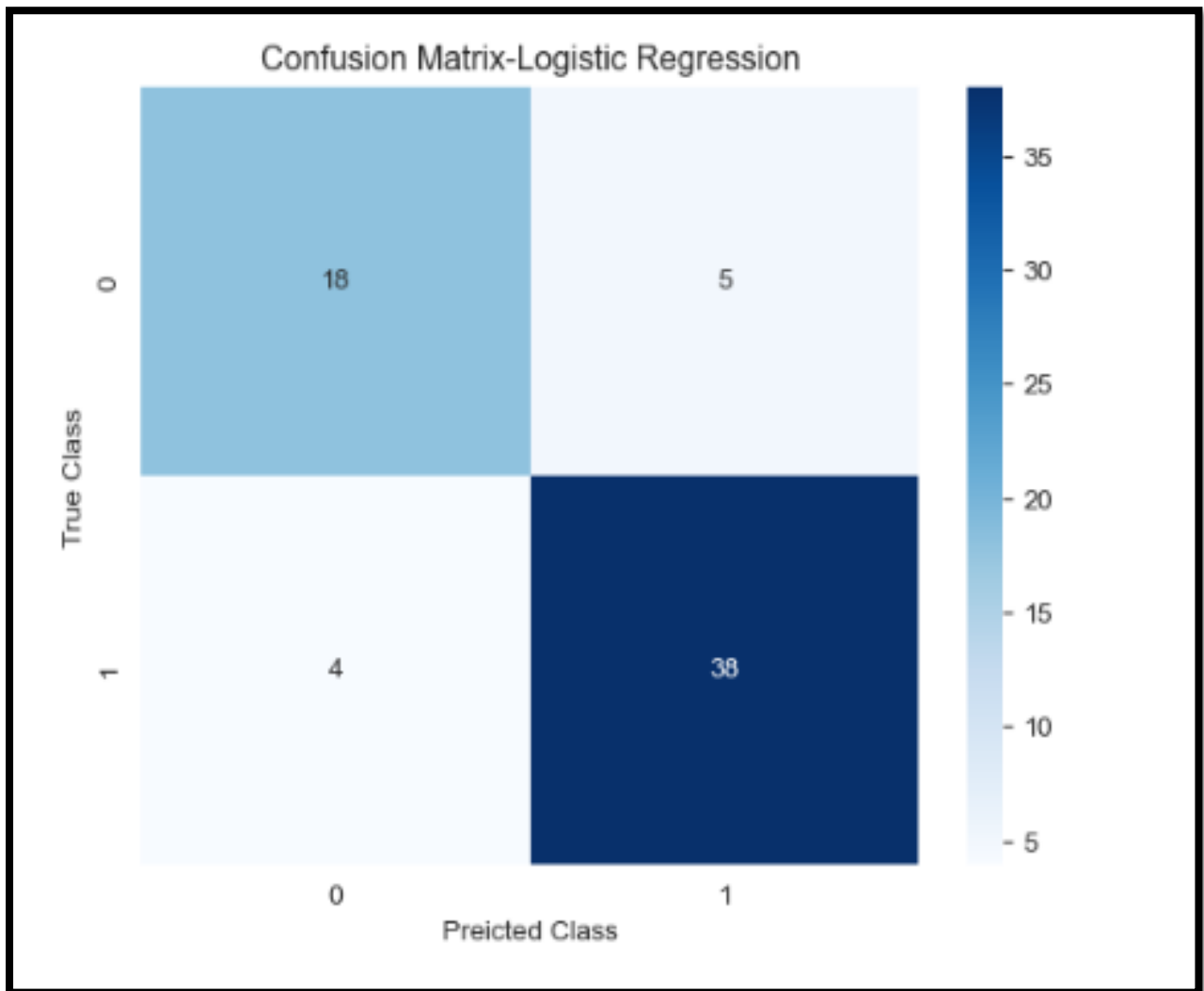
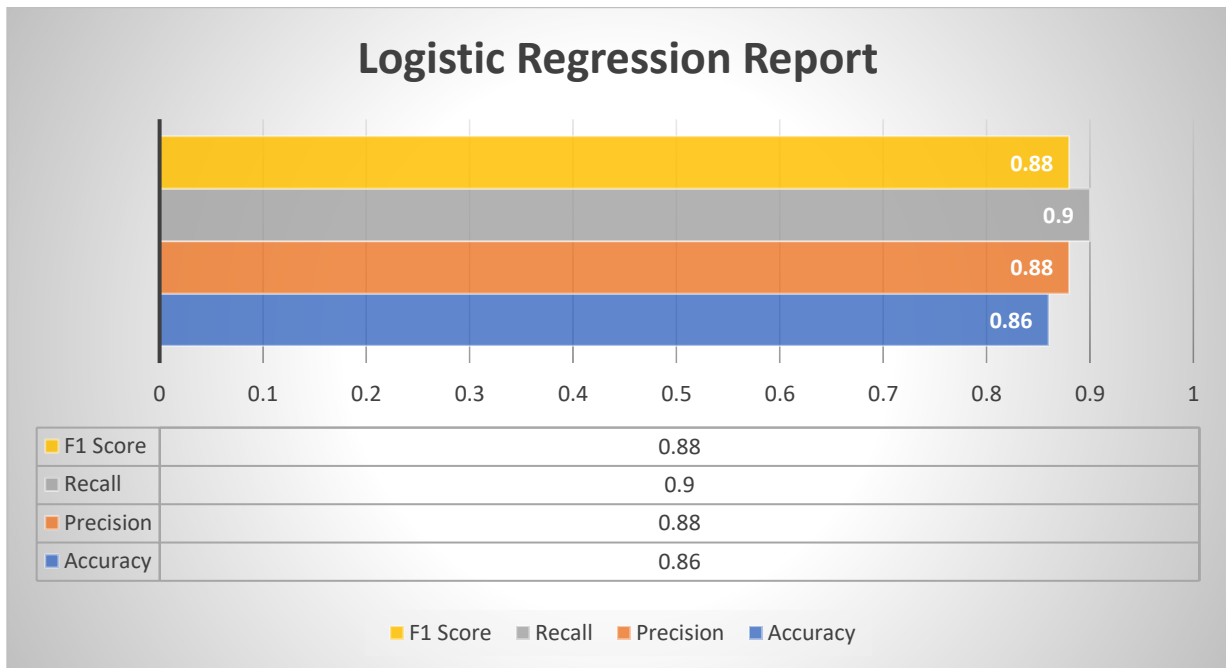


Figure 35 Confusion Matrix of Logistic Regression





*Table 1 Logistic Regression Graphical Performance*

### 6.1.3 Decision Tree Classification

The following steps have been taken to perform the Decision Tree Classification.

- a. Importing different libraries for data extraction, data visualization, and machine learning algorithms
- b. These libraries are:
  - i. Pandas Library
  - ii. NumPy Library
  - iii. Seaborn Library
  - iv. Matplotlib.pyplot Library
  - v. TensorFlow Library

After importing the aforementioned libraries in the Jupyter Notebook, programs were made to analyze, visualize, and predict the data using the Decision Tree Algorithm.

```
In [74]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

```
In [75]: from sklearn.tree import DecisionTreeClassifier
```

```
In [76]: from sklearn.metrics import classification_report,confusion_matrix
```

```
In [77]: dtree = DecisionTreeClassifier()
```

```
In [78]: dtree.fit(X_train,y_train)
```

```
Out[78]: ▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

```
In [79]: predictions = dtree.predict(X_test)
```

```
In [80]: from sklearn.metrics import classification_report,confusion_matrix
```

```
In [81]: print(classification_report(y_test,predictions))
```

...

```
In [82]: print(confusion_matrix(y_test,predictions))
```

*Figure 36 Python Code of Decision Tree Classification*

### 6.1.4 Decision Tree Results

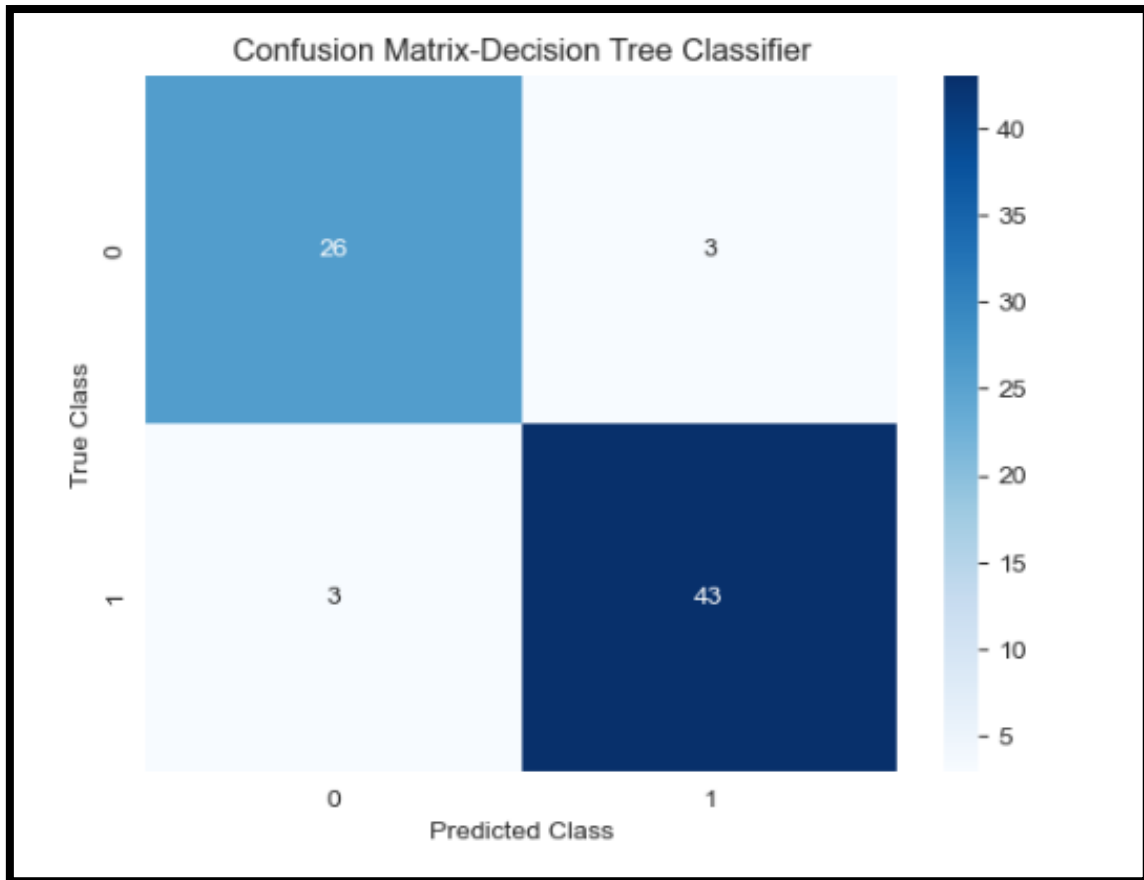


Figure 37 Decision Tree Confusion Matrix

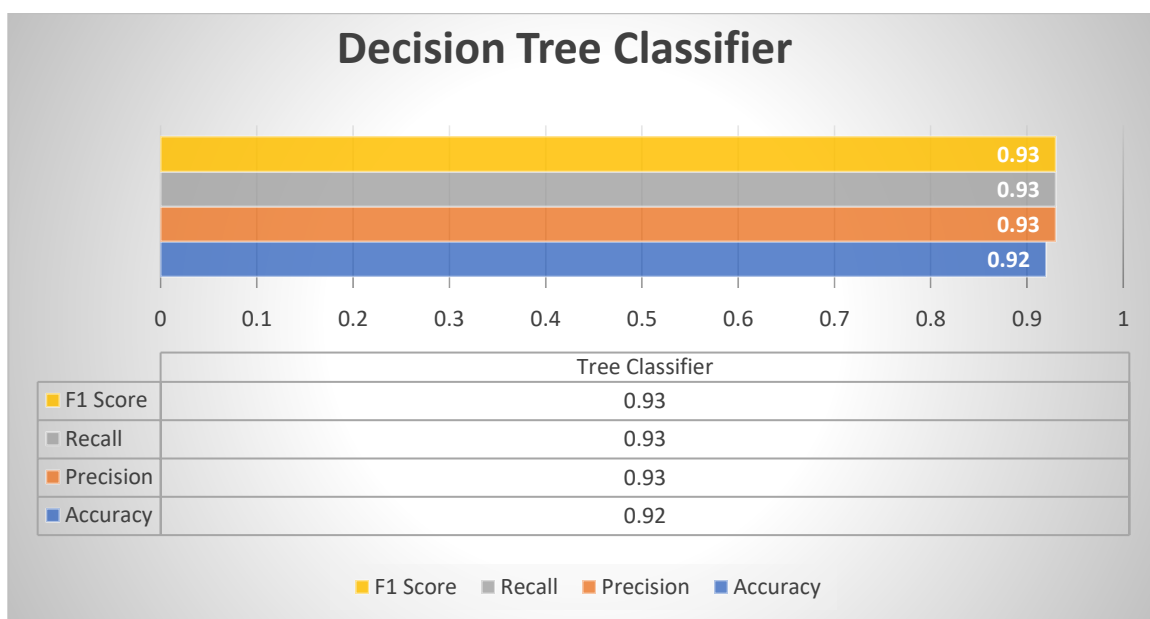


Table 2 Tree Classifier Graphical Performance

### 6.1.5 Support Vector Machine Classification

The following steps have been taken to perform the Support Vector Machine Classification.

- a. Importing different libraries for data extraction, data visualization, and machine learning algorithms
- b. These libraries are:
  - i. Pandas Library
  - ii. NumPy Library
  - iii. Seaborn Library
  - iv. Matplotlib.pyplot Library
  - v. TensorFlow Library

After importing the aforementioned libraries in the Jupyter Notebook, programs were made to analyze, visualize, and predict the data using the Support Vector Machine.

```
In [106]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

```
In [107]: from sklearn.svm import SVC
```

```
In [108]: model = SVC()
```

```
In [109]: model.fit(X_train,y_train)
```

```
Out[109]: SVC
SVC()
```

```
In [110]: predictions = model.predict(X_test)
```

```
In [111]: from sklearn.metrics import classification_report,confusion_matrix
```

```
In [112]: print(confusion_matrix(y_test,predictions))
```

...

```
In [113]: print(classification_report(y_test,predictions))
```

Figure 38 Support Vector Machine Python Code

6.1.6 Support Vector Machine Results

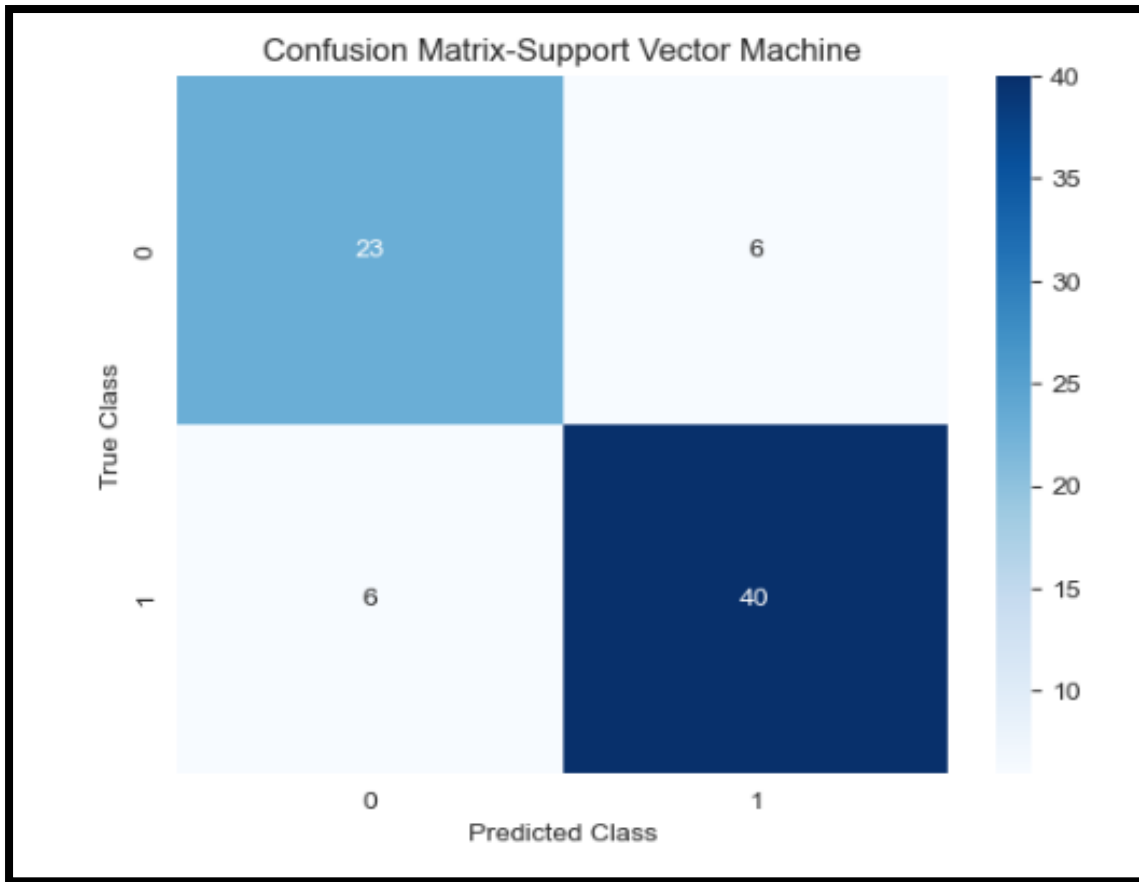


Figure 39 Support Vector Machine Confusion Matrix

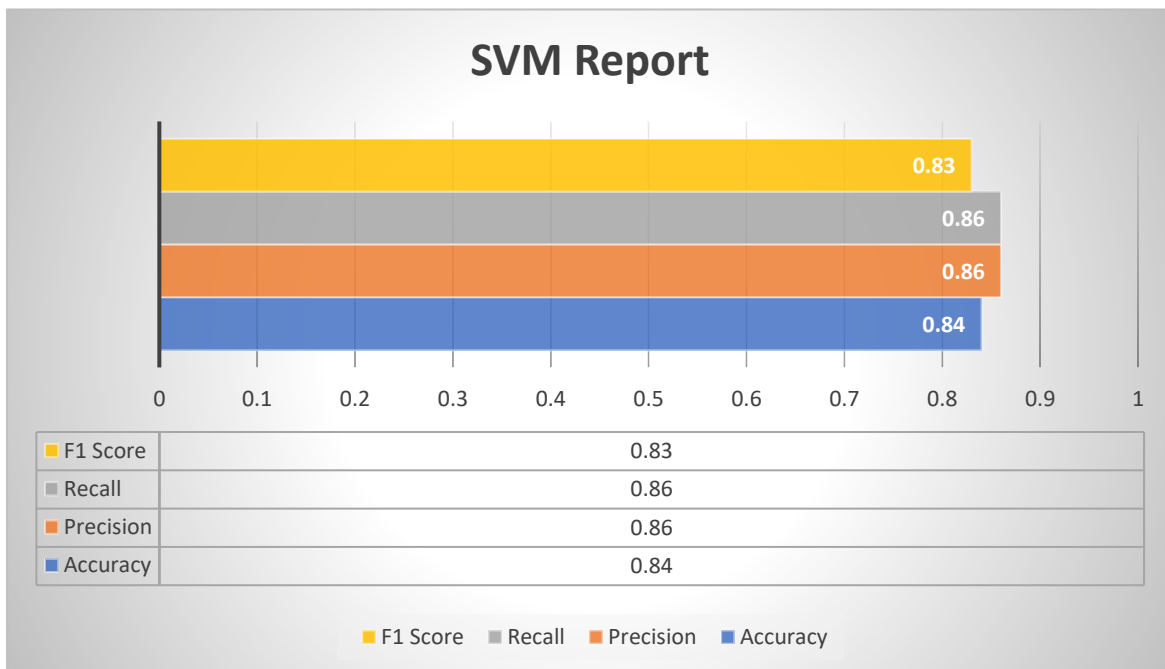


Table 3 Support Vector Machine Graphical Performance

### 6.1.7 Artificial Neural Network Classification

The following steps have been taken to perform the Artificial Neural Network.

- a. Importing different libraries for data extraction, data visualization, and machine learning algorithms
- b. These libraries are:
  - i. Pandas Library
  - ii. NumPy Library
  - iii. Seaborn Library
  - iv. Matplotlib.pyplot Library
  - v. TensorFlow Library

After importing the aforementioned libraries in the Jupyter Notebook, programs were made to analyze, visualize, and predict the data using the Artificial Neural Network.

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

```
In [2]: pip install tensorflow
```

```
In [53]: df=pd.read_csv('Tip resistance data.csv')
```

```
In [54]: sns.countplot(x='Liquefaction', data=df)
```

...

```
In [5]: df.corr()
```

```
In [6]: sns.heatmap(df.corr(), cmap='coolwarm', annot=True)
plt.title('Correlation Matrix')
```

```
In [7]: df.corr()['Liquefaction'].sort_values
```

```
In [8]: df.corr()['Liquefaction'].sort_values().plot(kind='bar')
plt.title('Correlation by Bar Graph')
```

```
In [9]: X = df.drop('Liquefaction',axis=1).values
y = df['Liquefaction'].values
```

```
In [10]: from sklearn.model_selection import train_test_split
```

```
In [11]: X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.25,random_state=101)
```

```
In [12]: from sklearn.preprocessing import MinMaxScaler
```

```

In [13]: scaler = MinMaxScaler()

In [14]: scaler.fit(X_train)

In [15]: X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)

In [16]: import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation, Dropout

In [17]: X_train.shape

In [18]: model = Sequential()

In [19]: model.compile(loss='binary_crossentropy', optimizer='adam')

In [20]: model.fit(x=X_train,
                  y=y_train,
                  epochs=600,
                  validation_data=(X_test, y_test), verbose=1
                  )

In [21]: model.history.history

In [22]: model_loss = pd.DataFrame(model.history.history)

In [23]: model_loss.plot()

In [24]: model = Sequential()
model.add(Dense(units=30,activation='relu'))
model.add(Dense(units=15,activation='relu'))
model.add(Dense(units=1,activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam')

In [25]: from tensorflow.keras.callbacks import EarlyStopping

In [26]: early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=25)

In [27]: model.fit(x=X_train,
                  y=y_train,
                  epochs=600,
                  validation_data=(X_test, y_test), verbose=1,
                  callbacks=[early_stop]
                  )

In [28]: model_loss = pd.DataFrame(model.history.history)
model_loss.plot()

In [29]: from tensorflow.keras.layers import Dropout

In [30]: model = Sequential()
model.add(Dense(units=30,activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(units=15,activation='relu'))
model.add(Dropout(0.5))

model.add(Dense(units=1,activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam')

In [31]: model.fit(x=X_train,
                  y=y_train,
                  epochs=600,
                  validation_data=(X_test, y_test), verbose=1,
                  callbacks=[early_stop]

```

Figure 40 Artificial Neural Network Python Code

### 6.1.8 Artificial Neural Network Results

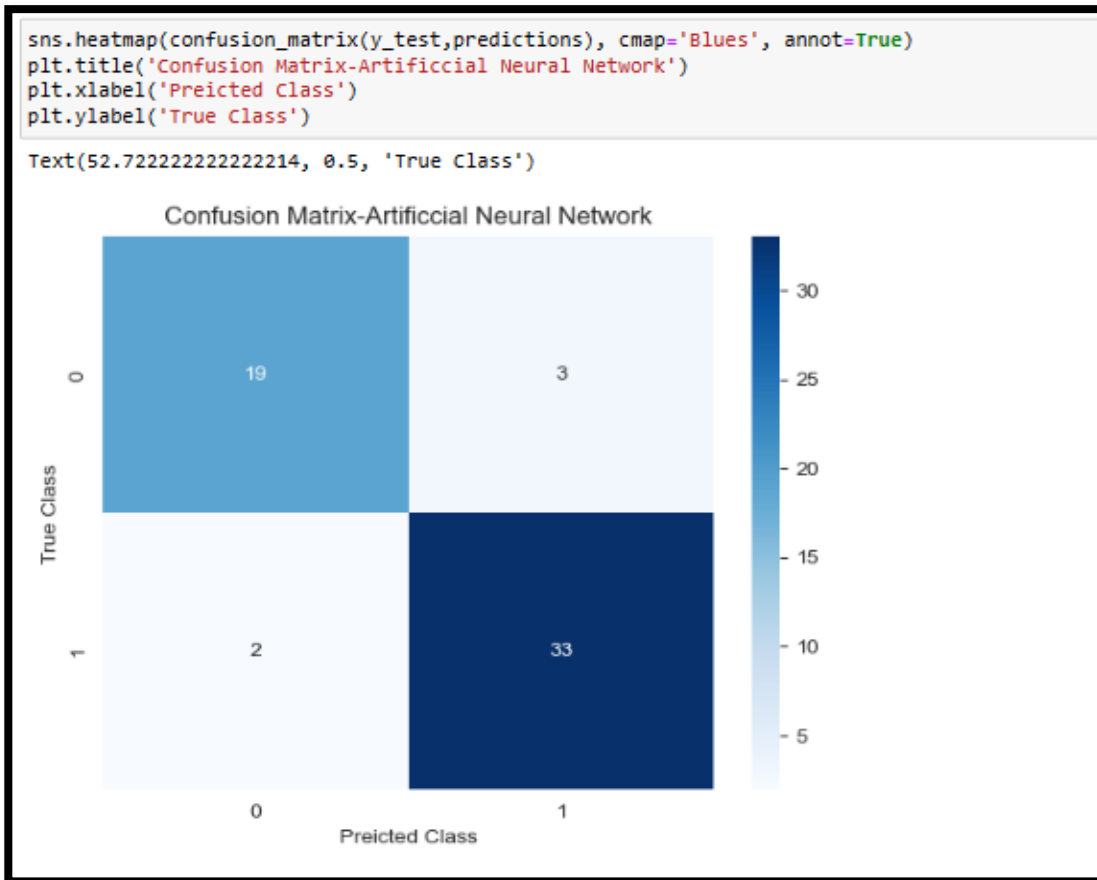


Figure 41 Confusion Matrix of ANN

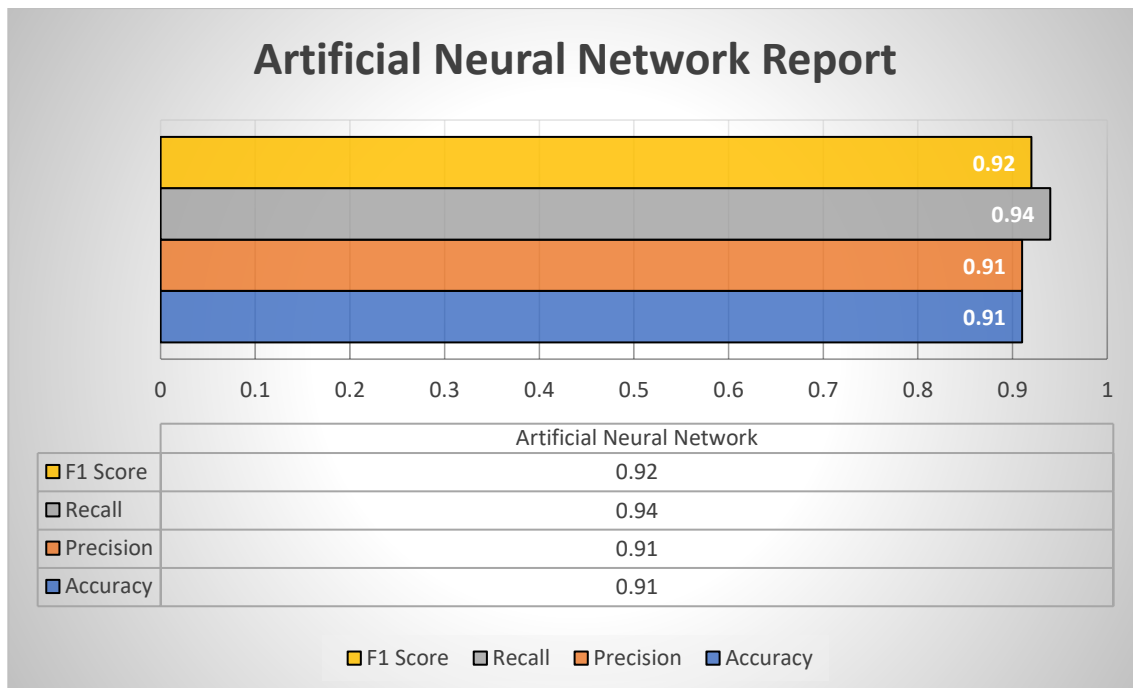


Table 4 Artificial Neural Network Graphical Performance



## 6.2 Discussion

- **Logistic Regression:**

|           |      |
|-----------|------|
| Accuracy  | 0.86 |
| Precision | 0.88 |
| Recall    | 0.90 |
| F1 Score  | 0.88 |

*Table 5 Logistic Regression Report*

As has been discussed, Logistic Regression has a limitation when there are outliers. Some outliers were removed using the box plot programming; however, completely eradicating all the outliers would have decreased our data to a large extent. Therefore, removing all the outliers is recommended when the datasets are in multiple thousands. In the domain of geotechnical engineering, data collection is an arduous and hectic process, and generating thousands of sets of data is improbable.

Logistic regression is an indispensable tool in the machine learning toolkit box, particularly when interpretation and efficiency are imperative. However, its linear nature and assumptions inhibit its applicability in cases involving deep and complex relationships, non-linear patterns, and high-dimensional data. As always, the choice of algorithm should depend on the characteristics of the data and the goals of the task at hand

- **Support Vector Machines:**

|           |      |
|-----------|------|
| Accuracy  | 0.84 |
| Precision | 0.86 |
| Recall    | 0.86 |
| F1 Score  | 0.83 |

*Table 6 Support Vector Machine Report*

Support Vector Machines are powerful classifiers that can be utilized in various scenarios. However, their effectiveness is closely influenced by careful hyperparameter tuning and the choice of appropriate kernel functions might not be the best choice for extremely large datasets or situations where interpretability and probabilistic outputs are critical. Like any machine learning algorithm, understanding the strengths and limitations of SVMs is crucial for making informed decisions about their usage.

Support Vector Machines aim to find the hyperplane that best separates the classes while maximizing the margin. However, they can be sensitive to noise and outliers in data. Outliers can significantly influence the position and orientation of the optimal hyperplane, leading to suboptimal classification results. This is the reason SVM has been unable to give the best predictions.

- **Artificial Neural Network and Decision Trees**

|           |      |
|-----------|------|
| Accuracy  | 0.91 |
| Precision | 0.91 |
| Recall    | 0.94 |
| F1 Score  | 0.92 |

*Table 7 Artificial Neural Network Report*

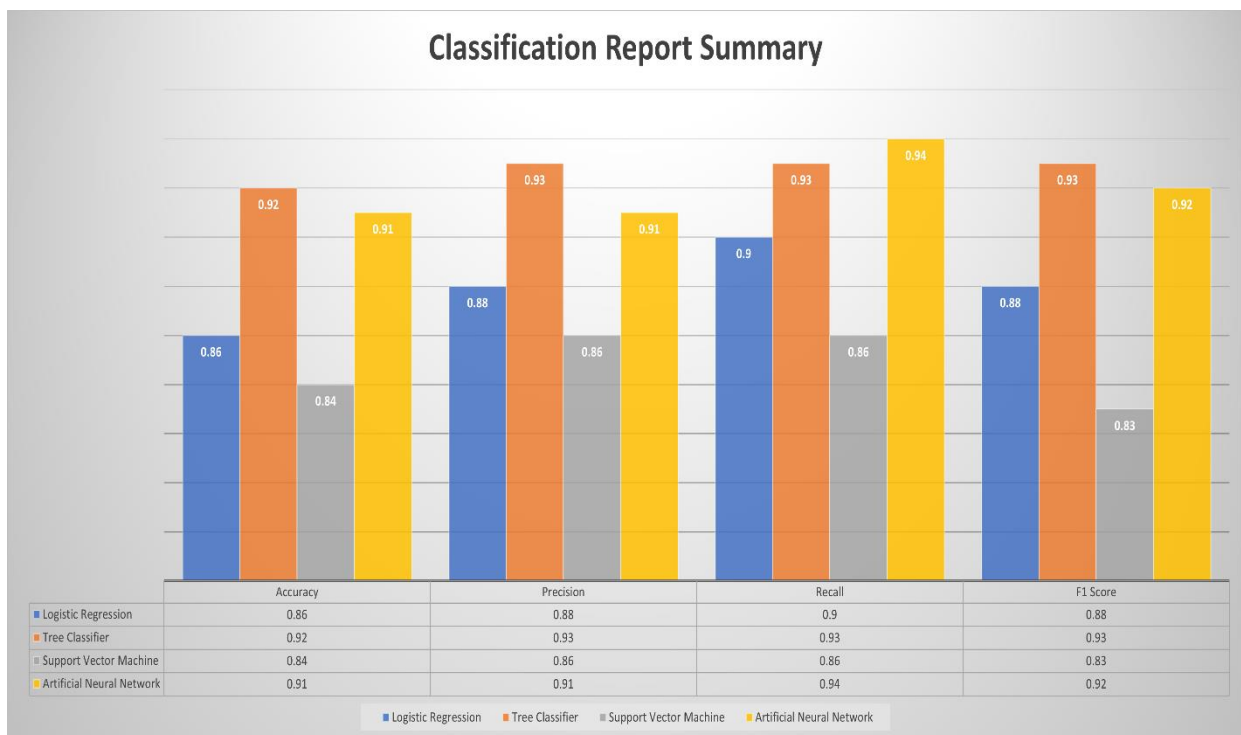
|           |      |
|-----------|------|
| Accuracy  | 0.92 |
| Precision | 0.93 |
| Recall    | 0.93 |
| F1 Score  | 0.93 |

*Table 8 Decision Tree Report*

The performance parameters of different algorithms have been gauged using Python language and Jupyter Notebook. A comparative analysis of these algorithms' final report shows that *Artificial Neural Networks and Decision Tree* performance were the best among all four algorithms. However, ANN performed better than Decision Tree. Neural Networks perform better when there are multiple hidden layers, and in our case, the data was not sufficient to provide

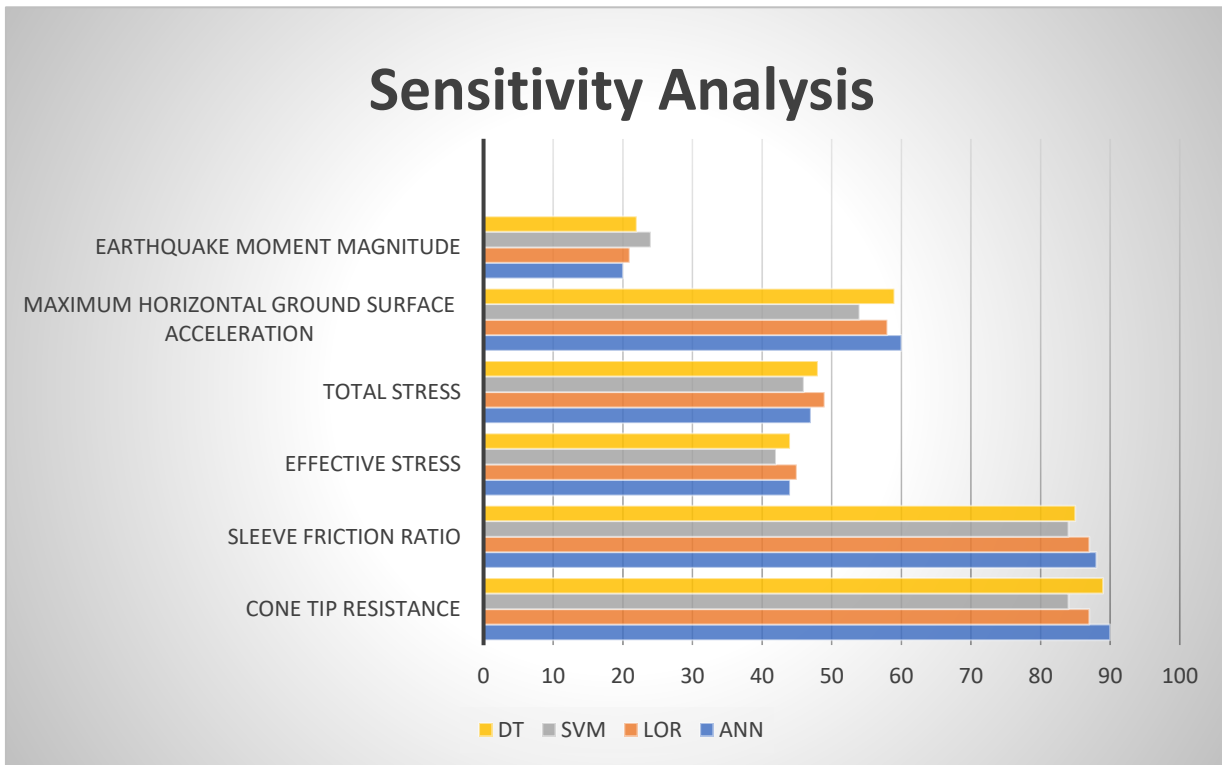
hidden layers. Had multiple hidden layers been used in the algorithm, it would have overfit the algorithm. Furthermore, the predictions would have vacillated a lot due to the overfitting of the model. Because an overfit model has high bias and low variance, it is always suggested that a model should not overfit the data.

Artificial Neural Networks are adept at modeling intricate patterns and high-dimensional data but require substantial data and computational resources. Decision Trees are interpretable, efficient, and useful for tasks with clear feature importance, but they can overpower and struggle with non-linear connections. The choice between ANNs and Decision Trees depends on factors such as the complexity of the problem, the available data, the need for interpretability, and the resources available for training and inference



*Figure 42 Summary Report of All Algorithms used*

- **Sensitivity Analysis**



*Table 9 Sensitivity Analysis*

- Sensitivity Analysis shows that Cone penetration resistance is the most important input parameter in the data set.
- After CPT, different input parameters have sensitivity in the following decreasing order:
  - Sleeve Friction Ratio
  - Maximum horizontal ground surface acceleration
  - Total Stress
  - Effective Stress
  - Earthquake Magnitude

### 6.3 Practical Demonstration of the Algorithm

| Test Data            |                       |
|----------------------|-----------------------|
| Tip Resistance       | 8 MPa                 |
| Sleeve Friction      | 2 %                   |
| Effective Stress     | 50 kPa                |
| Total Stress         | 100 kPa               |
| Maximum Acceleration | 0.36 m/s <sup>2</sup> |
| Moment Magnitude     | 7.1                   |

Table 10 Test Data with Tip Resistance 8 MPa

```
In [71]: df_pre=pd.DataFrame(np.array([[8,2,50,100,0.36,7.1]]),
                             columns=['qc_Mpa','R f %','eff_stress','total_stress','amax','Mw'])

In [72]: predictions = logmodel.predict(X_test)

In [73]: predict=logmodel.predict(df_pre)

In [74]: predict
Out[74]: array([0], dtype=int64)

In [75]: if predict == [0]:
          print('Liquefaction is not susceptible')
          elif predict == [1]:
          print('Liquefaction is susceptible')

          Liquefaction is not susceptible

In [76]: probability=logmodel.predict_proba(df_pre)

In [77]: probability
Out[77]: array([[0.88869654, 0.11130346]])
```

The new test data has been used to predict the liquefaction susceptibility in terms of the probability of an event. Tip resistance with a value of 8 MPa was used as a variable input parameter to gauge the effect on the liquefaction susceptibility while keeping all the input parameters constant. After the algorithm was fed with the above-mentioned values, it showed that there are 88 % chance that the soil with these input parameters will not liquefy

| Test Data            |                       |
|----------------------|-----------------------|
| Tip Resistance       | 6 MPa                 |
| Sleeve Friction      | 2 %                   |
| Effective Stress     | 50 kPa                |
| Total Stress         | 100 kPa               |
| Maximum Acceleration | 0.36 m/s <sup>2</sup> |
| Moment Magnitude     | 7.1                   |

Table 11 Test Data with Tip Resistance 6 MPa

```

In [78]: df_pre=pd.DataFrame(np.array([[6,2,50,100,0.36,7.1]]),
                             columns=['qc_Mpa','R f %','eff_stress','total_stress','amax','Mw'])

In [79]: predictions = logmodel.predict(X_test)

In [80]: predict=logmodel.predict(df_pre)

In [81]: predict
Out[81]: array([0], dtype=int64)

In [82]: if predict == [0]:
           print( 'Liquefaction is not susceptible')
         elif predict == [1]:
           print('Liquefaction is susceptible')

           Liquefaction is not susceptible

In [83]: probability=logmodel.predict_proba(df_pre)

In [84]: probability
Out[84]: array([[0.72985764, 0.27014236]])

```

When the value of tip resistance is decreased from 8 MPa to 6MPa keeping all the other values constant, the algorithm gave the output that there is a 72 % chance that Liquefaction is not susceptible. It shows that the probability of no liquefaction decreased from 88 percent to 72 %.

| Test Data       |       |
|-----------------|-------|
| Tip Resistance  | 4 MPa |
| Sleeve Friction | 2 %   |

|                      |                       |
|----------------------|-----------------------|
| Effective Stress     | 50 kPa                |
| Total Stress         | 100 kPa               |
| Maximum Acceleration | 0.36 m/s <sup>2</sup> |
| Moment Magnitude     | 7.1                   |

Table 12 Test Data with Tip Resistance 4 MPa

```

In [57]: df_pre=pd.DataFrame(np.array([[4,2,50,100,0.36,7.1]]),
                             columns=['qc_Mpa','R f %','eff_stress','total_stress','amax','Mw'])

In [58]: predictions = logmodel.predict(X_test)

In [59]: predict=logmodel.predict(df_pre)

In [60]: predict
Out[60]: array([1], dtype=int64)

In [61]: if predict == [0]:
           print('Liquefaction is not susceptible')
         elif predict == [1]:
           print('Liquefaction is susceptible')

           Liquefaction is susceptible

In [62]: probability=logmodel.predict_proba(df_pre)

In [63]: probability
Out[63]: array([[0.47759138, 0.52240862]])

```

Similarly, when the value of Cone Penetration resistance was further decreased to 4MPa while keeping all the input parameters constant, the algorithm suggested that there are 53 % chance that the soil would liquefy under the given circumstances. So, in this way, these machine learning models could be used in practical fieldwork to measure and estimate the liquefaction susceptibility of the soil.

## ***Chapter. 7***

### **7. CONCLUSION AND FUTURE RECOMMENDATIONS**

#### **7.1 Conclusion**

With the technological advancement in every domain of the field, practitioners are shifting their inclinations and energies towards soft computations which require less energy and effort to save time and money, and at the same time give results that surpass the old computational and obsolete methods. In this study, an effort has been made to predict the liquefaction potential of liquefiable soils using four different machine-learning algorithms. As liquefaction assessment is a laborious task in the field as well as in the laboratory, it requires extreme human force, effort, and financial resources to perform field and laboratory tests. Furthermore, these tests have inherent limitations due to the generalizations made for the formulation of different assessment formulas.

Being cognizant of the limitations of simple techniques and approaches, the engineering world is rapidly changing and transmogrifying the old techniques with new ones which are cost-effective, and accurate. In this research four supervised machine learning algorithms are used to predict the liquefaction potential. These algorithms are Logistic Regression, Support Vector Machines, Decision Tree Classifiers, and Artificial Neural Networks.

The database was collected using the previous research papers and literature where supervised and unsupervised algorithms were applied to the data. The input parameters used to develop the training model were Cone tip resistance, sleeve friction ratio, effective stress, total stress, maximum horizontal ground surface acceleration, and earthquake moment magnitude. The dataset consisted of 226 instances of liquefaction and non-liquefaction case histories. In the development of all the models, 70 percent of the data has been used for the training of the model,



and the remaining 30 percent of the data is used for the testing of the algorithms. Liquefaction, as a probabilistic output, has been the output of the algorithms as a binary classification problem. Four different performance criteria were used in the research to ascertain the performance of the generated models. These include Accuracy, Precision, Recall, and F1 scores.

- It was observed that two models, namely Decision Tree and Artificial Neural Network performed the best among all the algorithms.
- The performance of the Artificial Neural Network was better than the performance of the Decision Tree in this study.
- In the case of highly inseparable data, as in the case of this study, Logistic Regression and Support Vector Machines do not perform well because they work best when the data is linearly separable.
- The performance criteria defined by Accuracy can be misleading as it has certain limitations. To counter these limitations, three other performance criteria were used for the assessment. It has been observed that all the performance criteria were giving more or less similar results, which is indicative of a good model generation.
- If a large database is available to the engineers, Artificial Neural Networks perform the best. However, in case of paucity of the dataset, the Decision Tree algorithm should be used for the development of the prediction model.
- An imbalanced data set should not be used for the development of a model as in that case the prediction model will sway towards the most likely situation and overfit the one particular situation.
- Cone Penetration Resistance has the most significant impact on the overall performance of the model. It should be noted that CPT values are properly documented and do not

contain any null values. However, if the values are zero in some dataset, it should be replaced by the mean of the particular column.

- The presence of an outlier, especially in an important variable, can significantly reduce the performance of the machine learning model. CPT or SPT data should be cleaned beforehand so that a model can best fit that accurately represents the whole dataset.

## 7.2 Recommendations

- In the future, different input parameters can be introduced like the fine content of the soil, its water content, etc. to predict the liquefaction potential.
- Use of ensemble learning is a new technique that uses different base classifiers to predict the output. Artificial Neural Networks and Tree Classifiers can be used as base models in the future to further refine the output accuracy of the algorithms.
- Feature engineering is a technique where new features are introduced by manipulating and performing mathematical functions on the dataset. It should be applied to the dataset to extract the best features that can give the best results.
- Convolution Neural Networks learn on visual data to predict the output. In the future, CNN can be used to train the model by the incorporation of images of the soils that liquefy under given loading conditions. This can further increase the prediction capability of the Neural Networks.
- Liquefaction has been treated by practitioners as a binary classification due to its nature. Binary classification has some limitations in terms of labeling outputs as 0 and 1. These absolute values can make it difficult to interpret the probability of any event. To counter this issue in the future, probabilistic models can be introduced based on Bayes Theorem that can give the probability score of the liquefaction potential assessment

## References

- [1] Abbaszadeh Shahri, A. (2016). Assessment and Prediction of Liquefaction Potential Using Different Artificial Neural Network Models: A Case Study. *Geotechnical and Geological Engineering*, 34(3), 807–815. <https://doi.org/10.1007/s10706-016-0004-z>
- [2] Ahmad, M., Tang, X.-W., Qiu, J.-N., Ahmad, F., & Gu, W.-J. (2021). Application of machine learning algorithms for the evaluation of seismic soil liquefaction potential. *Frontiers of Structural and Civil Engineering*, 15(2), 490–505. <https://doi.org/10.1007/s11709-020-0669-5>
- [3] Baziar, M. H., & Nilipour, N. (2003). Evaluation of liquefaction potential using neural networks and CPT results. *Soil Dynamics and Earthquake Engineering*, 23(7), 631–636. [https://doi.org/10.1016/S0267-7261\(03\)00068-X](https://doi.org/10.1016/S0267-7261(03)00068-X)
- [4] Chern, S.-G., Lee, C.-Y., & Wang, C.-C. (2008). CPT-BASED LIQUEFACTION ASSESSMENT BY USING FUZZY-NEURAL NETWORK. *Journal of Marine Science and Technology*, 16(2). <https://doi.org/10.51400/2709-6998.2024>
- [5] Erzin, Y., & Ecemis, N. (2015). The use of neural networks for CPT-based liquefaction screening. *Bulletin of Engineering Geology and the Environment*, 74(1), 103–116. <https://doi.org/10.1007/s10064-014-0606-8>
- [6] Galupino, J., & Dungca, J. (2023). Estimating Liquefaction Susceptibility Using Machine Learning Algorithms with a Case of Metro Manila, Philippines. *Applied Sciences*, 13(11), 6549. <https://doi.org/10.3390/app13116549>
- [7] Goh, A. T. C., & Goh, S. H. (2007). Support vector machines: Their use in geotechnical engineering as illustrated using seismic liquefaction data. *Computers and Geotechnics*, 34(5), 410–421. <https://doi.org/10.1016/j.compgeo.2007.06.001>
- [8] Hanna, A. M., Ural, D., & Saygili, G. (2007). A neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data. *Soil Dynamics and Earthquake Engineering*, 27(6), 521–540. <https://doi.org/10.1016/j.soildyn.2006.11.001>
- [9] Hanzawa, H. (1979). Undrained Strength Characteristics of an Alluvial Marine Clay in the Tokyo Bay. *Soils and Foundations*, 19(4), 69–84. [https://doi.org/10.3208/sandf1972.19.4\\_69](https://doi.org/10.3208/sandf1972.19.4_69)
- [10] Hoang, N.-D., & Bui, D. T. (2018). Predicting earthquake-induced soil liquefaction based on a hybridization of kernel Fisher discriminant analysis and a least squares support vector machine: a multi-dataset study. *Bulletin of Engineering Geology and the Environment*, 77(1), 191–204. <https://doi.org/10.1007/s10064-016-0924-0>
- [11] Jas, K., & Dodagoudar, G. R. (2023). Explainable machine learning model for liquefaction potential assessment of soils using XGBoost-SHAP. *Soil Dynamics and Earthquake Engineering*, 165, 107662. <https://doi.org/10.1016/j.soildyn.2022.107662>
- [12] Juang, C. H., Yuan, H., Lee, D.-H., & Lin, P.-S. (2003). Simplified Cone Penetration Test-based Method for Evaluating Liquefaction Resistance of Soils. *Journal*

- of *Geotechnical and Geoenvironmental Engineering*, 129(1), 66–80. [https://doi.org/10.1061/\(ASCE\)1090-0241\(2003\)129:1\(66\)](https://doi.org/10.1061/(ASCE)1090-0241(2003)129:1(66))
- [13] Khandelwal, M., Marto, A., Fatemi, S. A., Ghorogi, M., Armaghani, D. J., Singh, T. N., & Tabrizi, O. (2018). Implementing an ANN model optimized by genetic algorithm for estimating the cohesion of limestone samples. *Engineering with Computers*, 34(2), 307–317. <https://doi.org/10.1007/s00366-017-0541-y>
- [14] Kohestani, V. R., Hassanlourad, M., & Ardakani, A. (2015). Evaluation of liquefaction potential based on CPT data using random forest. *Natural Hazards*, 79(2), 1079–1089. <https://doi.org/10.1007/s11069-015-1893-5>
- [15] Kramer, S. L., & Seed, H. B. (1988). Initiation of Soil Liquefaction Under Static Loading Conditions. *Journal of Geotechnical Engineering*, 114(4), 412–430. [https://doi.org/10.1061/\(ASCE\)0733-9410\(1988\)114:4\(412\)](https://doi.org/10.1061/(ASCE)0733-9410(1988)114:4(412))
- [16] Kumar, D., Samui, P., Kim, D., & Singh, A. (2021). A Novel Methodology to Classify Soil Liquefaction Using Deep Learning. *Geotechnical and Geological Engineering*, 39(2), 1049–1058. <https://doi.org/10.1007/s10706-020-01544-7>
- [17] Kurnaz, T. F., & Kaya, Y. (2019a). A novel ensemble model based on GMDH-type neural network for the prediction of CPT-based soil liquefaction. *Environmental Earth Sciences*, 78(11), 339. <https://doi.org/10.1007/s12665-019-8344-7>
- [18] Kurnaz, T. F., & Kaya, Y. (2019b). A novel ensemble model based on GMDH-type neural network for the prediction of CPT-based soil liquefaction. *Environmental Earth Sciences*, 78(11), 339. <https://doi.org/10.1007/s12665-019-8344-7>
- [19] Liu, J. (2020). Influence of Fines Contents on Soil Liquefaction Resistance in Cyclic Triaxial Test. *Geotechnical and Geological Engineering*, 38(5), 4735–4751. <https://doi.org/10.1007/s10706-020-01323-4>
- [20] Muduli, P. K., & Das, S. K. (2014). Evaluation of liquefaction potential of soil based on standard penetration test using multi-gene genetic programming model. *Acta Geophysica*, 62(3), 529–543. <https://doi.org/10.2478/s11600-013-0181-6>
- [21] Pal, M. (2006). Support vector machines-based modeling of seismic liquefaction potential. *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(10), 983–996. <https://doi.org/10.1002/nag.509>
- [22] Poulos, S. J., Castro, G., & France, J. W. (1985). Liquefaction Evaluation Procedure. *Journal of Geotechnical Engineering*, 111(6), 772–792. [https://doi.org/10.1061/\(ASCE\)0733-9410\(1985\)111:6\(772\)](https://doi.org/10.1061/(ASCE)0733-9410(1985)111:6(772))
- [23] Rahman, M. S., & Wang, J. (2002). Fuzzy neural network models for liquefaction prediction. *Soil Dynamics and Earthquake Engineering*, 22(8), 685–694. [https://doi.org/10.1016/S0267-7261\(02\)00059-3](https://doi.org/10.1016/S0267-7261(02)00059-3)
- [24] Ramakrishnan, D., Singh, T. N., Purwar, N., Barde, K. S., Gulati, Akshay., & Gupta, S. (2008). Artificial neural network and liquefaction susceptibility assessment: a case study using the 2001 Bhuj earthquake data, Gujarat, India. *Computational Geosciences*, 12(4), 491–501. <https://doi.org/10.1007/s10596-008-9088-8>
- [25] Rinne, E. E. (1987). *Assessing the Effects of Potential Liquefaction – a Practising Engineer’s Perspective* (pp. 245–251). <https://doi.org/10.1016/B978-0-444-98958-1.50021-2>
- [26] Samui, P., Kim, D., & Sitharam, T. G. (2011). Support vector machine for evaluating seismic-liquefaction potential using shear wave velocity. *Journal of Applied Geophysics*, 73(1), 8–15. <https://doi.org/10.1016/j.jappgeo.2010.10.005>

- [27] SHI, X., ZHOU, J., WU, B., HUANG, D., & WEI, W. (2012). Support vector machines approach to mean particle size of rock fragmentation due to bench blasting prediction. *Transactions of Nonferrous Metals Society of China*, 22(2), 432–441. [https://doi.org/10.1016/S1003-6326\(11\)61195-3](https://doi.org/10.1016/S1003-6326(11)61195-3)
- [28] Unno, T., Kazama, M., Uzuoka, R., & Sento, N. (2008). Liquefaction of Unsaturated Sand Considering the Pore Air Pressure and Volume Compressibility of the Soil Particle Skeleton. *Soils and Foundations*, 48(1), 87–99. <https://doi.org/10.3208/sandf.48.87>
- [29] Vaid, Y. P., Chern, J. C., & Tumi, H. (1985). Confining Pressure, Grain Angularity, and Liquefaction. *Journal of Geotechnical Engineering*, 111(10), 1229–1235. [https://doi.org/10.1061/\(ASCE\)0733-9410\(1985\)111:10\(1229\)](https://doi.org/10.1061/(ASCE)0733-9410(1985)111:10(1229))
- [30] Xue, X., & Yang, X. (2013). Application of the adaptive neuro-fuzzy inference system for prediction of soil liquefaction. *Natural Hazards*, 67(2), 901–917. <https://doi.org/10.1007/s11069-013-0615-0>
- [31] Youd, T. L., & Keefer, D. K. (1994). Liquefaction during the 1977 San Juan Province, Argentina earthquake (Ms = 7.4). *Engineering Geology*, 37(3–4), 211–233. [https://doi.org/10.1016/0013-7952\(94\)90057-4](https://doi.org/10.1016/0013-7952(94)90057-4)
- [32] Zhang, J., & Wang, Y. (2021). An ensemble method to improve prediction of earthquake-induced soil liquefaction: a multi-dataset study. *Neural Computing and Applications*, 33(5), 1533–1546. <https://doi.org/10.1007/s00521-020-05084-2>
- [33] Zhang, W., & Goh, A. T. C. (2018). Assessment of Soil Liquefaction Based on Capacity Energy Concept and Back-Propagation Neural Networks. In *Integrating Disaster Science and Management* (pp. 41–51). Elsevier. <https://doi.org/10.1016/B978-0-12-812056-9.00003-8>
- [34] Zhou, J., Huang, S., Wang, M., & Qiu, Y. (2022). Performance evaluation of hybrid GA–SVM and GWO–SVM models to predict earthquake-induced liquefaction potential of soil: a multi-dataset investigation. *Engineering with Computers*, 38(S5), 4197–4215. <https://doi.org/10.1007/s00366-021-01418-3>
- [35] Zhou, J., Li, E., Wang, M., Chen, X., Shi, X., & Jiang, L. (2019). Feasibility of Stochastic Gradient Boosting Approach for Evaluating Seismic Liquefaction Potential Based on SPT and CPT Case Histories. *Journal of Performance of Constructed Facilities*, 33(3). [https://doi.org/10.1061/\(ASCE\)CF.1943-5509.0001292](https://doi.org/10.1061/(ASCE)CF.1943-5509.0001292)
- [36] Zhou, J., Li, X., & Mitri, H. S. (2015). Comparative performance of six supervised learning methods for the development of models of hard rock pillar stability prediction. *Natural Hazards*, 79(1), 291–316. <https://doi.org/10.1007/s11069-015-1842-3>