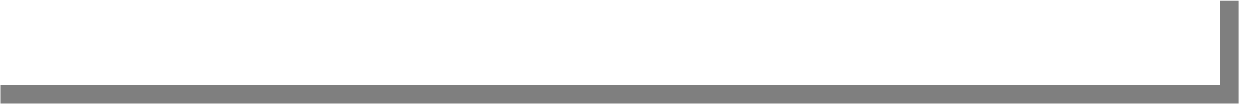
**1.**

A Project Report On



**EarthQuekes Dataset**

Submitted in partial fulfillment of the requirement for the award of the degree

MASTER OF COMPUTER APPLICATION

from

Marwadi University

Academic Year 2025 – 26

**Himanshu Jagatiya (92400584170)**

**Yash Chavda (92400584193)**

**Internal Guide**

**PROF. GEHNA SACHDEVA**



Rajkot-Morbi Road, At & PO : Gauridad, Rajkot 360 003. Gujarat. India.



**Faculty of Computer Applications (FoCA)**

****

**This is to certify that the project work entitled Heart Failure Prediction Using Logistic Regression submitted in partial fulfillment of the requirement for**

**the award of the degree of**

Master of Computer Application

**of the**

Marwadi University

**is a result of the bonafide work carried out by**

**Himanshu Jagatiya(92400584170)**

**Yash Chavda(92400584193)**

**during the academic year 2025 – 2026**

|  |  |  |
| --- | --- | --- |
| *PROF. Gehna Sachdeva* | *Dr. Sunil Bajeja* | *Dr. R. Sridaran* |
| **Faculty Guide** | **HOD** | **Dean** |

# DECLARATION

We hereby declare that this project work entitled **EARTHQUEKE PREDICTION DATASET** is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place: Rajkot Date: 30Aug

Himanshu Jagatiya (92400584170) Signature:\_\_\_\_\_\_\_\_\_\_\_\_

Yash Chavda (92400584193) Signature:\_\_\_\_\_\_\_\_\_\_\_\_

# ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our guide **PROF. GEHNA SACHDEVA,** the person who makes us to follow the right steps during our project work. We express our deep sense of gratitude for her guidance, suggestions and expertise at every stage. A part from that her valuable and expertise suggestion during documentation of our report indeed help us a lot.

Thanks to our friend and colleague who have been a source of inspiration and motivation that helped to us during our project work.

We are heartily thankful to the Dean of our department **Dr. R. Sridaran** for giving us an opportunity to work over this project and for their end-less and great support to all other people who directly or indirectly supported and help us to fulfil our task.

Himanshu Jagatiya (92400584170) Signature:\_\_\_\_\_\_\_\_\_\_\_\_

Yash Chavda (92400584193) Signature:\_\_\_\_\_\_\_\_\_\_\_\_

## CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapters** | **Particulars** | **Page No.** |
| 1 | **Introduction**   * 1. Objective of the New System   2. Problem Definition   3. Core Components   4. Project Profile   5. Assumptions and Constraints   6. Advantages and Limitations of the Proposed System | 6 |
| 2 | **Requirement Determination & Analysis**   * 1. Requirement Determination   2. Targeted Users   3. Tool details (Python / PowerBI/ Tableau)   4. Library description (Details on various libraries / packages used) | 7 |
| 3 | **System Design**   * 1. Flowchart / Algorithm with steps   2. Dataset Design   3. Details on preprocessing steps applied | 8 |
| 4 | **Development**  4.1 Script details / Source code  4.2. Screen Shots / UI Design of simulation (if applicable)  4.3. Test reports | 10 |
| 5 | **Proposed Enhancements** | 20 |
| 6 | **Conclusion** | 21 |
| 7 | **Bibliography** | 21 |

1. **Introduction:**

### Objective of the New System:

The primary objective is to create a robust system for analyzing the provided earthquake data. This involves developing a user-friendly interface to query, visualize, and identify significant patterns in seismic events. The system will leverage this data to build predictive models, aiming to forecast potential seismic activity. Ultimately, it seeks to transform raw data into actionable insights for research and safety purposes.

### Problem Definition:

The core problem is the overwhelming volume and complexity of seismic data, making manual analysis impractical. It's difficult to identify meaningful trends, correlations, and precursor signals hidden within the vast dataset. This lack of efficient analysis tools hinders timely risk assessment and scientific understanding. The new system is intended to automate this process, addressing the challenge of extracting critical information from extensive data.

### Core Components:

* + - **Dataset:** Earthquekes predictive dataset
    - **Preprocessing:** Handling null values, encoding, scaling
    - **Algorithms:** Logistic Regression and comparison with KNN, SVM, DT, RF, NB
    - **Evaluation:** Accuracy, Confusion Matrix, Graphs

### Project Profile:

* + - **Project Title:** Earthquekes Prediction Using Classification
    - **Domain:** Healthcare Data Science
    - **Technology:** Python (sklearn, pandas, seaborn, matplotlib)
    - **Dataset:** Synthetic patient data with health attributes

### Assumptions and Constraints:

* + - **Assumptions:**
      * The dataset is clean after preprocessing.
      * Models are evaluated using accuracy and classification reports.

### Constraints:

* + - * Dataset may not represent real-world noise.
      * Imbalanced data may affect results.

### Advantages and Limitations of the Proposed System:

* + - **Advantages:**
      * Early prediction of tsunami and earthqueke risk.
      * Helps reduce human error in diagnosis.
      * Multiple algorithm comparison.

### Limitations:

* + - * Synthetic dataset might not reflect real conditions.
      * Feature engineering is limited.

## Requirement Determination & Analysis:

### Requirement Determination:

To build a predictive system for earthquake risk, the following requirements were identified:

* + - A dataset containing earthqueke metrics
    - Data preprocessing pipeline
    - Machine learning algorithms
    - Model evaluation techniques
    - Visualization tools for understanding the data

### Targeted Users:

The targeted users for this system are weather professionals, data scientists, and researchers who aim to predict earthqueke risk using state data.

### Tool details (Python / PowerBI/ Tableau):

This project is developed using Python due to its wide range of libraries and ease of implementation for data analysis and machine learning.

### Library description (Details on various libraries / packages used):

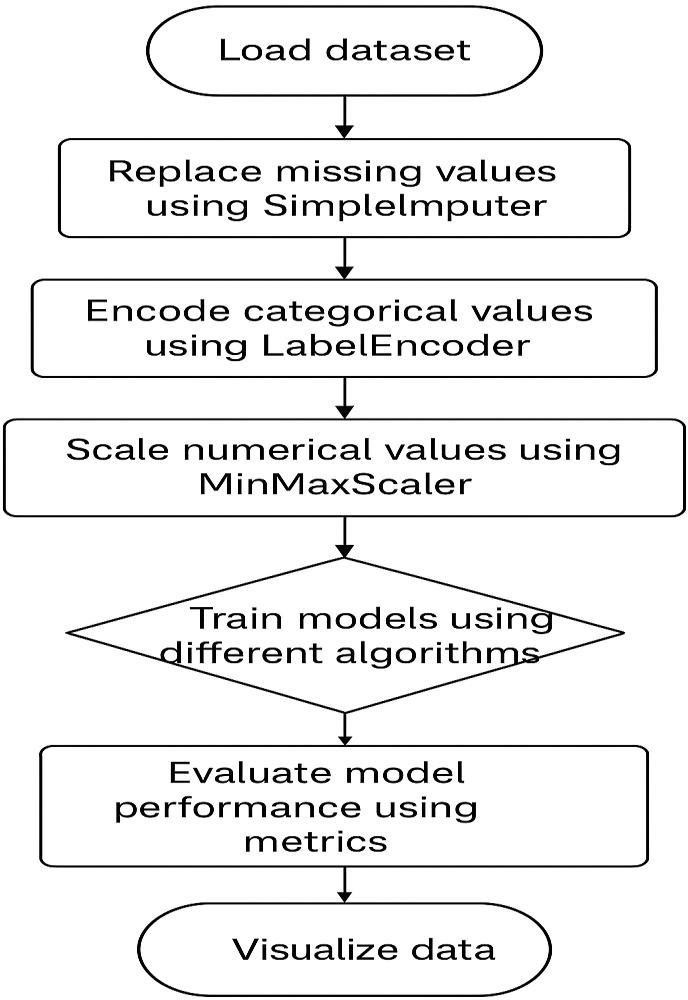
* + - **pandas:** Data manipulation and analysis
    - **numpy:** Numerical operations
    - **sklearn:** Machine learning algorithms and preprocessing tools
    - **seaborn:** Statistical data visualization
    - **matplotlib:** Plotting graphs

## System Design:

### Flowchart / Algorithm with steps:

* **Algorithm:**
* Step 1: Load dataset
* Step 2: Replace missing values using SimpleImputer
* Step 3: Encode categorical values using LabelEncoder
* Step 4: Scale numerical values using MinMaxScaler
* Step 5: Split data into training and test sets
* Step 6: Train models using different algorithms
* Step 7: Evaluate model performance using metrics
* Step 8: Visualize data using heatmap, scatter, boxplot, and histogram

### Flowchart:

****

* 1. **Dataset Design:**



The dataset includes 30,000 synthetic samples with various attributes such as time, place, status, tsunami, significations, datatype, magnitude, state, longitude, latitude, depth and date. The target variable is "".

### Details on preprocessing steps applied:

* + - **Missing Values:** Filled using SimpleImputer (mean for numerical, most frequent for categorical)
    - **Categorical Encoding:** LabelEncoder applied to all object type columns
    - **Feature Scaling:** MinMaxScaler used on numerical columns to normalize them between 0 and 1.

## Development:

### Script Details / Source Code:

The project is divided into two main Python scripts:

1. **preprocessing.py:** Responsible for cleaning the dataset using SimpleImputer, encoding with LabelEncoder, and scaling using MinMaxScaler.

import pandas as pd

import numpy as np

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import MinMaxScaler, LabelEncoder

def preprocess\_data(file\_path):

"""Loads dataset, fills missing values, encodes categorical features, and scales numerical features."""

# Load Dataset

df = pd.read\_csv("/Earth90-23\_modified.csv")

# Replace '?' with NaN if present

df.replace('?', np.nan, inplace=True)

# Separate numeric and categorical columns

numerical\_cols = df.select\_dtypes(include=['number']).columns

categorical\_cols = df.select\_dtypes(include=['object']).columns

# === Handling Missing Values ===

# Apply SimpleImputer for numeric columns (mean strategy)

numeric\_imputer = SimpleImputer(strategy='mean')

df[numerical\_cols] = numeric\_imputer.fit\_transform(df[numerical\_cols])

# Apply SimpleImputer for categorical columns (most frequent strategy)

categorical\_imputer = SimpleImputer(strategy='most\_frequent')

df[categorical\_cols] = categorical\_imputer.fit\_transform(df[categorical\_cols])

# === Encoding Categorical Columns ===

label\_encoders = {}

for col in categorical\_cols:

le = LabelEncoder()

df[col] = le.fit\_transform(df[col])

label\_encoders[col] = le

# === Scaling Numerical Features ===

scaler = MinMaxScaler()

df[numerical\_cols] = scaler.fit\_transform(df[numerical\_cols])

return df, numerical\_cols, categorical\_cols, label\_encoders

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

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

file\_path = "/Earth90-23\_modified.csv"

df, numerical\_cols, categorical\_cols, label\_encoders = preprocess\_data(file\_path)

print(" Data Preprocessing Completed!")

print(df.head())

OUTPUT:-

Data Preprocessing Completed!

7.163110e+11 22km ENE of Barstow, California reviewed 0.0 92.0 1

7.163120e+11 13km NNW of Joshua Tree, California reviewed 0.0 87.0 1

35km ENE of Lucerne Valley, California reviewed 0.0 89.0 1

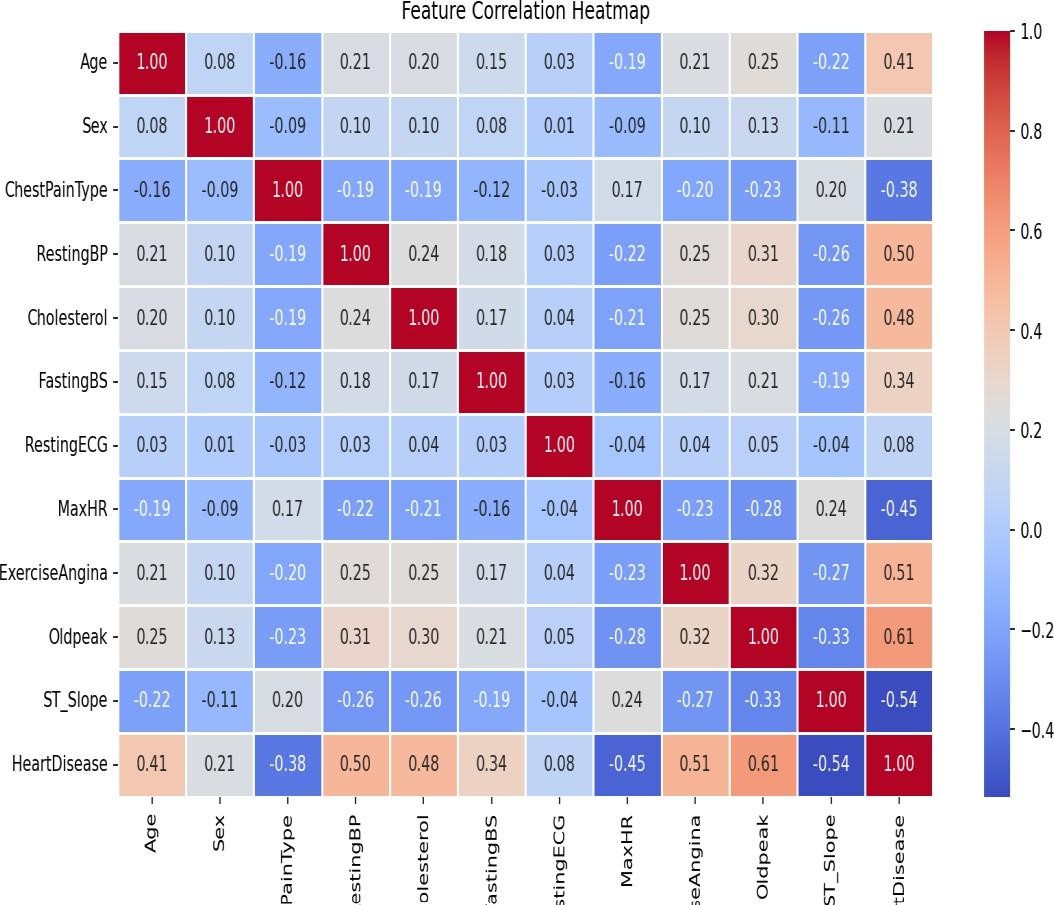
40km NE of Big Bear City, California reviewed 1.0 35.0 1

19km N of Barstow, California reviewed 0.0 122.0 1

### Graphical Visualization (heatmap, scatter plot, histogram, box plot):

The following visualizations were used:

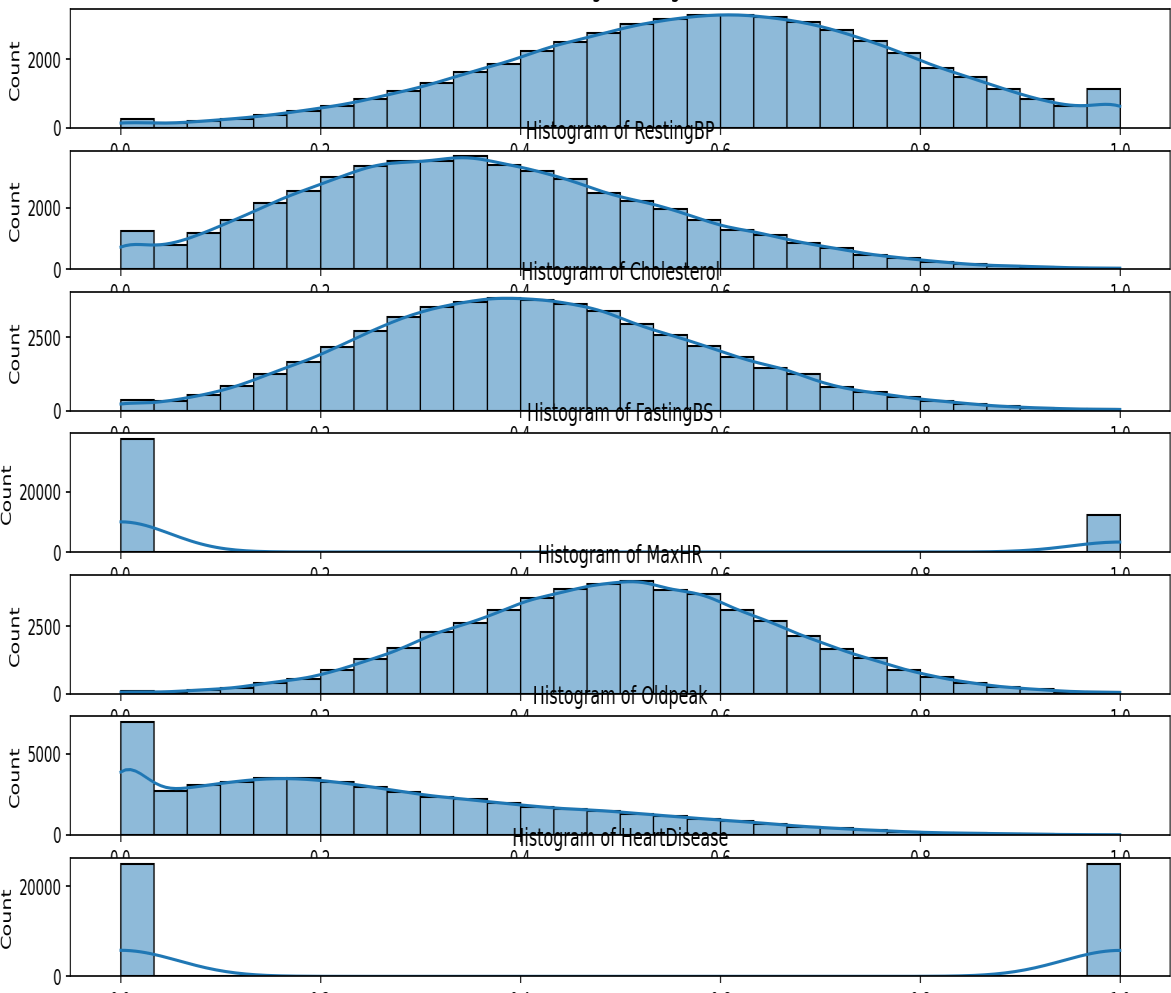
* + - **Heatmap:** Shows correlation between all features to identify the most influential variables.



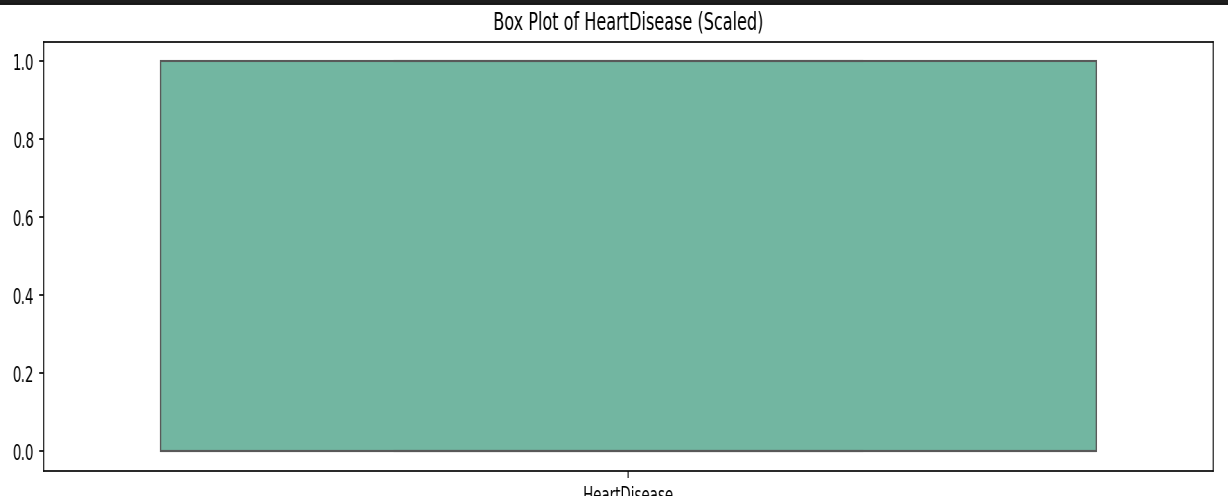
* + - **Scatter Plot:** Visualizes pairwise relationships between numerical features.

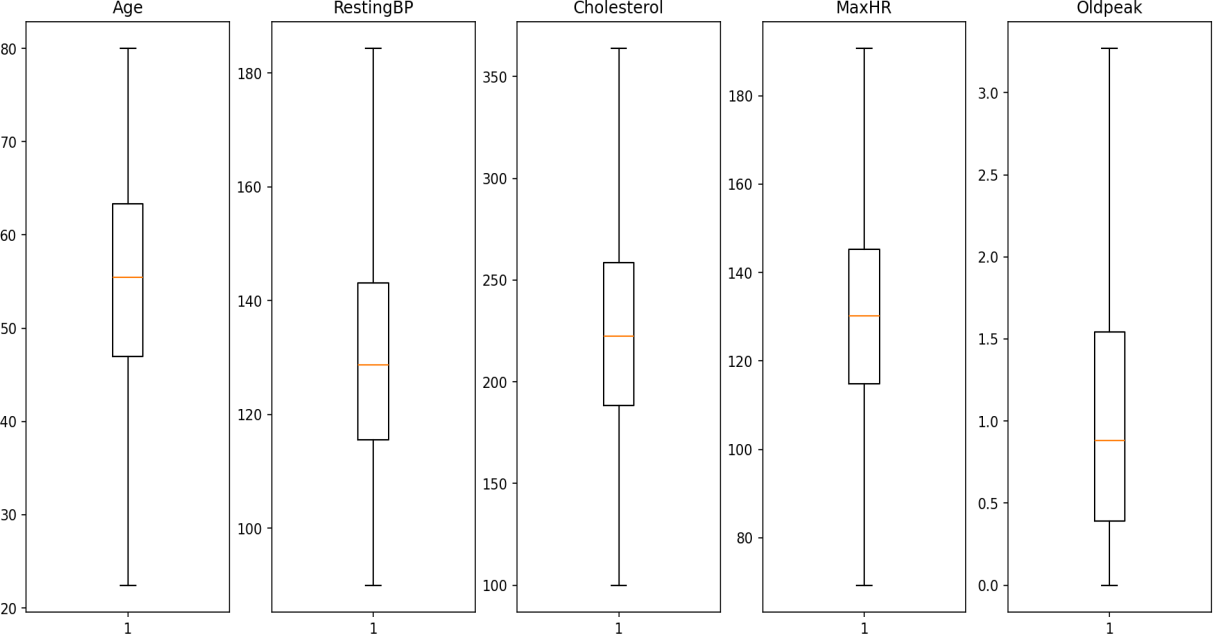


* + - **Histogram:** Displays the distribution of individual numerical features.



* + - **Box Plot:** Compares distribution of features based on heart disease status.





### Test Reports (Accuracy, Classification Reports):

Each model was evaluated using Confusion Matrix, Accuracy Score, and Classification Report.

Results include precision, recall, F1-score, and overall accuracy. Below are some sample findings:

* + - **Logistic Regression:** Accuracy – 97.33 %
    - **Random Forest:** Accuracy - 97.31 %
    - **Support Vector Machine:** Accuracy (kernel=poly) - 96.26 %
    - **K-Nearest Neighbors:** Accuracy improved with value of k (k=2) - 97.25%
    - **Naïve Bayes:** Fast and simple model with slightly lower accuracy -97.33 %
    - **Decision Tree:** Provided interpretability with moderate accuracy - 94.23 %

## Proposed Enhancements:

* Use of real-time whether data instead of synthetic data.
* Inclusion of additional features like date, states, earrhqueke history.
* Model deployment as a web or mobile app for IMD department.
* Integration of deep learning models for better accuracy.
* Use of ensemble models for higher robustness.

## Conclusion:

This project successfully demonstrated the implementation of a machine learning pipeline to predict tusnami using Logistic Regression. Multiple classifiers were evaluated and compared. The system provides a valuable foundation for further enhancements using real- world tsunami data and model deployment in a clinical setting.

## Bibliography:

1. scikit-learn: Machine Learning in Python – <https://scikit-learn.org/>
2. pandas Documentation – <https://pandas.pydata.org/>
3. seaborn Documentation – <https://seaborn.pydata.org/>
4. matplotlib Documentation – <https://matplotlib.org/>
5. Heart Disease Dataset (original inspiration) – <https://www.kaggle.com/>