CHAPTER 1
INTRODUCTION

1.1 General Introduction:

India has a 1.27 billion population, which is second-most in the entire world. It is the seventh-largest country in the world with an area of 3.288 million sq km. Indians are very much dependent on agriculture. It is the largest source of livelihood in India. In rural households, 70% of people are primarily dependent on agriculture, with about 82% of farmers being small and marginal. In 2020-21, total food grain production was estimated at 308.65 million tonnes (MT). India is the largest producer (25% of global production), the consumer (27% of world consumption), and the importer (14%) of pulses in the world.

India's annual milk production was 165 MT (2017-18), making India the largest producer of milk, jute, and pulses, with the world's second largest cattle population of 190 million in 2012. With merely 2.4% arable land resources and 4% water resources, Indian agriculture is feeding nearly 1.3 billion people, which implicates huge pressure on land and other natural resources for continuous productivity.

After the green revolution (which started in the 1960s), India made significant progress in agriculture production, which became possible due to modernization. With the development in technology, farmers have been provided with advanced farming techniques, better seeds (High Yielding Variety (HYV) seeds), mechanized farm tools, chemical fertilizers, facilities of irrigation, and electrical energy.

Since the green revolution, there has been excessive use of chemical fertilizers which has increased the crop productivity manifold. However, it has turned into a

problem as overuse of these chemical fertilizers has been detrimental for crop productivity and soil fertility. Fertilizer recommendations rarely match soil needs which has caused overuse of these chemical products.

So, there is a need for accurate fertilizer recommendations for the farmer and accurately analysing soil properties is the first step for that. Indian Agricultural Research Institute (ICAR) recommends soil test-based, balanced and integrated nutrient management through conjunctive use of both inorganic and organic sources of plant nutrients to reduce the use of chemical fertilizers, preventing deterioration of soil health, environment and contamination of groundwater.

This work aims to use various machine learning techniques to accurately predict the soil properties relevant for agriculture using spectroscopy data. Over the last 20 years, soil spectroscopy has become a powerful technique for analysing relative to the traditionally used chemical methods, particularly in the infrared range. Spectroscopy is known as a fast, economical, quantitative, and eco-friendly technique, which can be used in the fields as well as in the laboratory to provide hyper spectral data with narrow and numerous data.

Objectives:

The main objectives of this project are

- To predict or detect the soil properties effectively.
- To Predict the quality of the soil.
- To implement the machine learning algorithms.
- To enhance the overall performance for classification algorithms.

CHAPTER 2

SYSTEM STUDY

2.1 EXISTING SYSTEM:

In existing system, information about soil properties help the farmers to do effective and efficient farming, and yield more crops with less usage of resources. An attempt has been made in this project to predict the soil properties using machine learning approaches. The main properties of soil prediction are Calcium, Phosphorus, pH, Soil Organic Carbon, and Sand. These properties greatly affect the production of crops. Four well-known machine learning models, namely, multiple linear regression, random forest regression, support vector machine, and gradient boosting, are used for prediction of these soil properties. The performance of these models is evaluated on Africa Soil Property Prediction dataset. Experimental results reveal that the gradient boosting outperforms the other models in terms of coefficient of determination. Gradient boosting is able to predict all the soil properties accurately except phosphorus.

2.1.1 ISSUES:

- Doesn’t Efficient for handling large volume of data.
- Incorrect Classification Results.
- Less Prediction Accuracy.
- It doesn’t recommend the crop based on predicting soil.

2.1 PROPOSED SYSTEM:

In this system, the soil properties dataset as input was taken from dataset repository like UCI repository. Then, it implements the pre-processing step. In this step, it checks the missing values for wrong prediction. Then, it split the dataset into

test which is used for predicting the model and train is used for evaluating the model. The dataset is split based on ratio. Then, it implements the different machine learning regression algorithm such as random forest regression and KNN regression for predicting the soil properties and quality of the crop. The experimental results shows that some performance metrics such as MAE and MSE are accurate.

2.2.1 ADVANTAGES:

- Time consumption is low.
- The result or prediction is effective.
- The data is implemented with removing the unwanted data.

2.3 LITERATURE SURVEY:

Literature survey 1:

Title: Prospective of Indian agriculture: highly vulnerable to huge unproductivity and unsustainability

Authors: Arun Kumar, Balkrishna S. Bhople and Anil Kumar

Year: 2020

Methodology:

Balanced and judicious fertilizer application rates were used till mid 1960s; however, due to the green revolution inorganic fertilizer consumption increased tremendously and reached a maximum of 18.07 million tonnes (mt) of nutrients in 2000; from then on, the nation is facing a gradual decrease in growth and productivity. Similar amounts, viz. 16–18 mt of nutrients was incorporated in the soil year after year . Most preferable cereal crops of India are wheat and rice, and it is evident from research that their cultivation mined huge quantity of nutrients from the soil. Recent studies showed that organic matter was depleted from nearly 3.7 M ha soil, and there was clear evidence for land degradation due to indiscriminate use of inorganic fertilizers and pesticide. Intensive and continuous use of inorganic fertilizers is one major cause for depletion of soil organic matter (SOM) and consequently nutrient immobilization. Moreover, in India, in general, blanket fertilizer recommendations are followed for N, P and K which rarely match soil fertility needs. Secondary and micronutrients are also often ignored in different cropping systems. Many studies report that the use of inorganic fertilizers has a suppressive impact on SOM mineralization. Soil carbon and nitrogen are indirectly linked biologically; quality and quantity of soil carbon improve soil microbial functions, abundance and diversity though impact of long-term fertilization on soil microorganism-mediated carbon mineralization is not studied extensively

Advantages:

The agricultural sector is of vital importance for the region. It is undergoing a process of transition to a market economy, with substantial changes in the social, legal, structural, productive and supply set-ups, as is the case with all other sectors of the economy.

Issues:

Large-scale Agriculture is all about massive production. This production requires farmers to import farm inputs such as fertilizers, machines, pesticides, and herbicides. This is because, with increased cultivation, the land eventually becomes degraded.

Literature survey 2:

Title: Soil Organic Carbon Content Prediction Using Soil-Reflected Spectra: A Comparison of Two Regression Methods

Authors: **Sharon Gomes Ribeiro**

Year: 2021

Methodology:

Quantifying the organic carbon content of soil over large areas is essential for characterising the soil and the effects of its management. However, analytical methods can be laborious and costly. Reflectance spectroscopy is a well-established and widespread method for estimating the chemical-element content of soils. The aim of this study was to estimate the soil organic carbon (SOC) content using hyperspectral remote sensing. The data were from soils from two localities in the semi-arid region of Brazil. The spectral reflectance factors of the collected soil samples were recorded at wavelengths ranging from 350–2500 nm. Pre-processing techniques were employed, including normalisation, Savitzky–Golay smoothing and first-order derivative analysis. The data (n = 65) were examined both jointly and by soil class, and subdivided into calibration and validation to independently assess the performance of the linear methods. Two multivariate models were calibrated using the SOC content estimated in the laboratory by principal component regression (PCR) and partial least squares regression (PLSR). The study showed significant success in predicting the SOC with transformed and untransformed data, yielding acceptable-to-excellent predictions (with the performance-to-deviation ratio ranging from 1.40–3.38). In general, the spectral reflectance factors of the soils decreased with the increasing levels of SOC.

Advantages:

Higher soil organic carbon promotes soil structure or tilth meaning there is greater physical stability. This improves soil aeration (oxygen in the soil) and water drainage and retention, and reduces the risk of erosion and nutrient leaching.

Issues:

10 percent organic matter is the goal for landscaping sites, while soil used for fill for construction purposes are often rejected if the organic matter exceeds 5 percent. An additional disadvantage of soil organic matter is that organic matter addition can result in a loss of available nitrogen in the soil.

Literature survey 3:

Title: Mineralogical composition and C/N contents in soil and water among betel vineyards of coastal Odisha, India

Authors: Biswajit Patra

Year: 2020

Methodology:

The *Piper betle* L. leaves and their significance were described in various ayurvedic studies of India and China for its diverse use in cultural practices and treatment of various health disorders. The leaves of *P. betle* were used as post-meal mouth freshener in India for centuries. However, it offers economic benefits to farmers of Coastal India at large. Betel leaves cultivated agricultural soils play a significant role for their mineralogical composition. So, this present study aimed to find out the soil physicochemical characteristics and C/N contents of Betel vineyards of coastal Odisha. The soil and water samples were collected from local varieties of *P. betle* L. cultivated vineyards of Balasore, Ganjam, and Puri districts of Odisha and investigated their mineralogical composition. The soil mineralogy plays crucial role to understand the soil–plant relations. The mineralogical and elemental composition of soil samples were carried out by using various techniques like Fourier transform infrared spectroscopy, X-ray diffraction (XRD) and energy-dispersive X-ray fluorescence, scanning electron microscopy attached with energy-dispersive X-ray system. CHNS analyzer for quantification of hydrogen, nitrogen, carbon, and sulfur content. H_2O_2 (30%)-treated soil is employed to eliminate the organic carbon from mass soil samples.

Advantages:

The composition of a mineral can be expressed as a chemical formula, which simply gives the proportions of the different elements and groups of elements in the mineral. The latter notion (groups of elements) comes into play for those minerals which have a restricted range of composition.

Issues:

Through in-field testing (such as rolling soil together to determine how well the soil sticks together) or through lab testing for properties such as pH levels, chemical composition, and other factors, the composition of soil can be easily determined.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

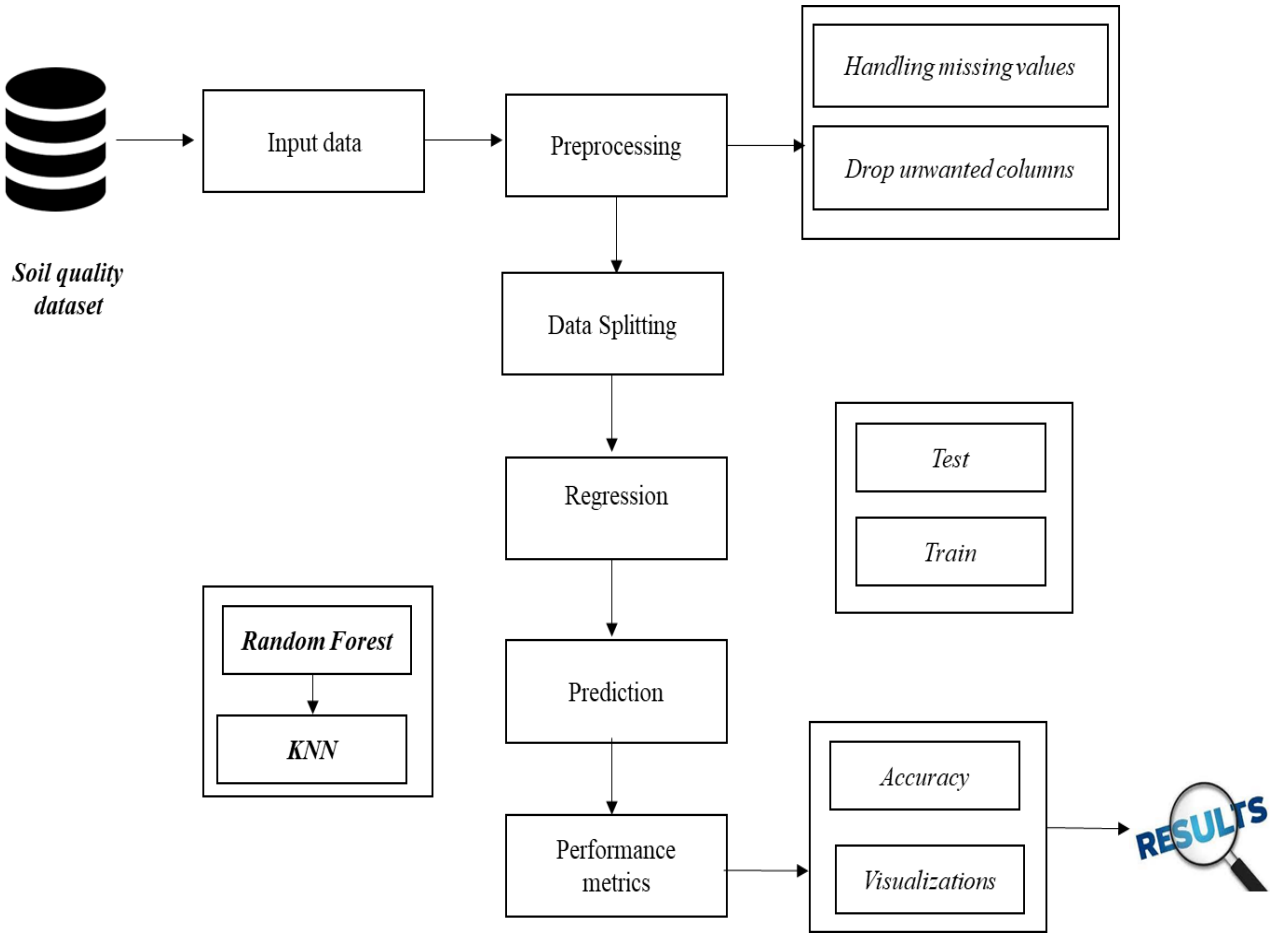


FIGURE 3.1: SYSTEM ARCHITECTURE

3.2 FLOW DIAGRAM

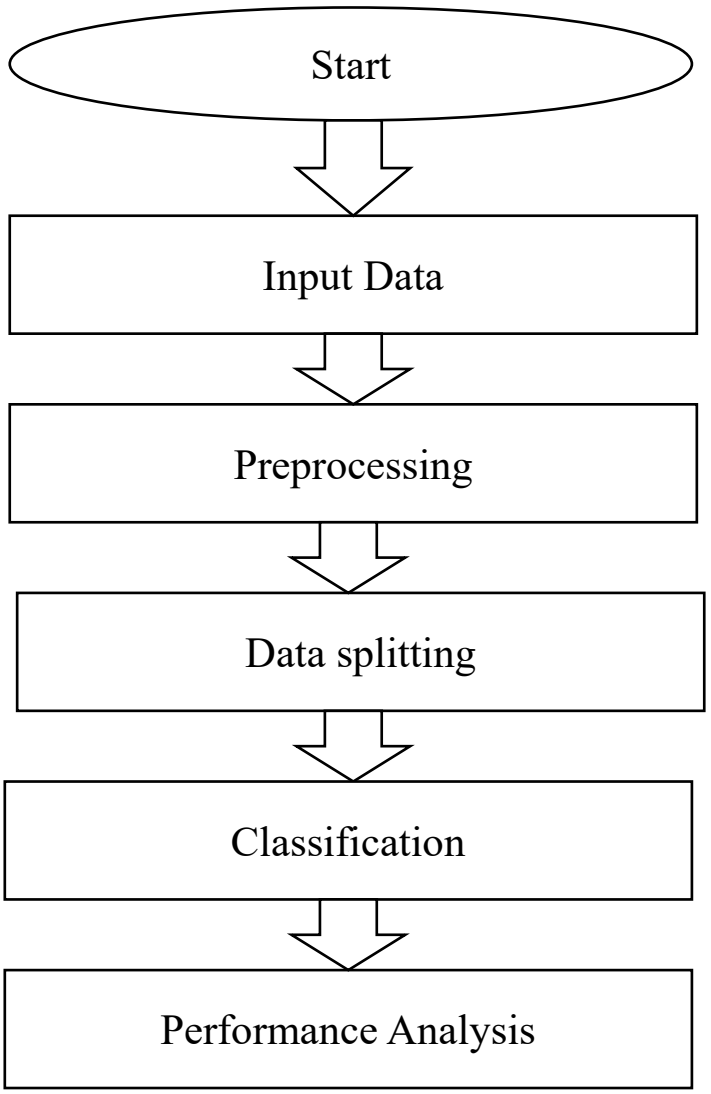


FIGURE 3.2: FLOW DIAGRAM

3.3 UML DIAGRAMS

3.3.1 USE CASE DIAGRAM

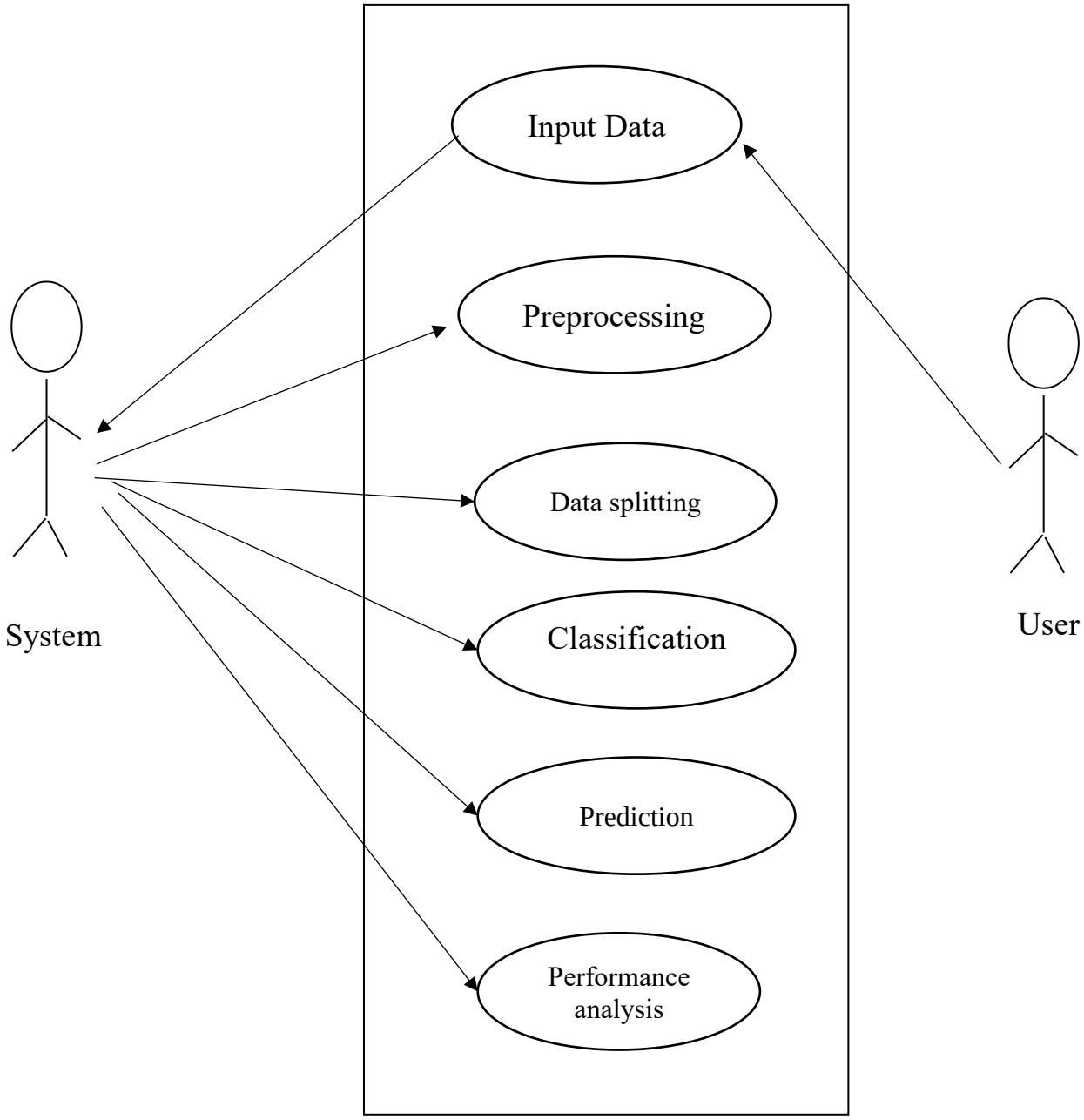


FIGURE 3.3.1: USE CASE DIAGRAM

3.3.2 USE CASE DIAGRAM

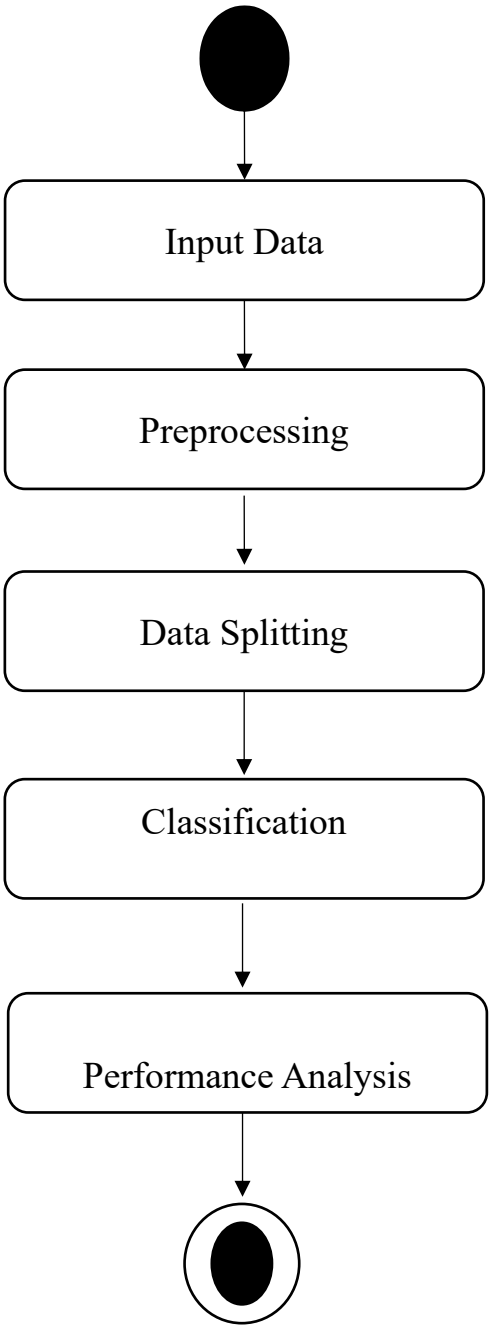


FIGURE 3.3.2: ACTIVITY CASE DIAGRAM

3.3.3 SEQUENCE DIAGRAM

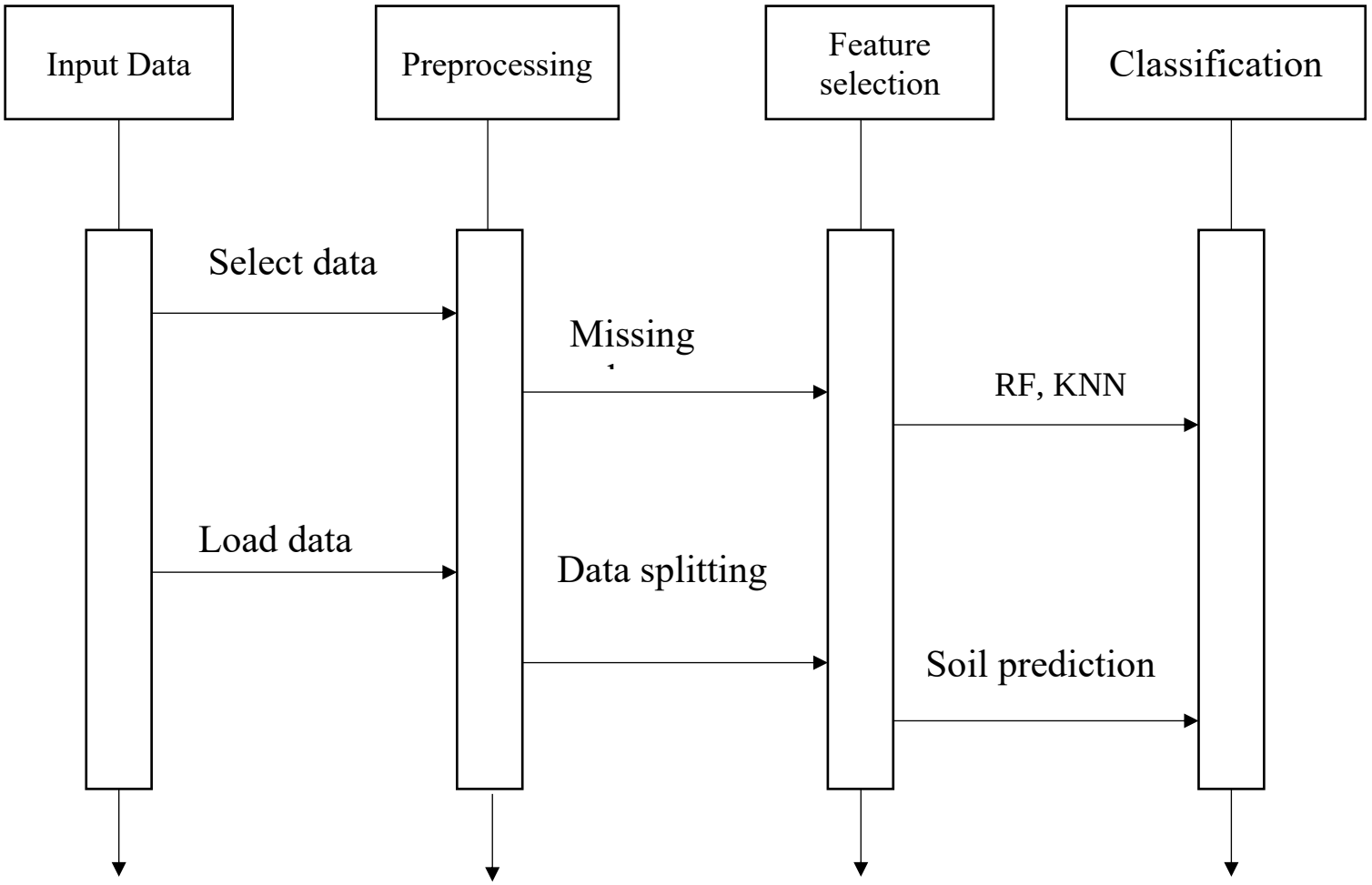


FIGURE 3.3.3: SEQUENCE DIAGRAM

3.3.4 ER DIAGRAM

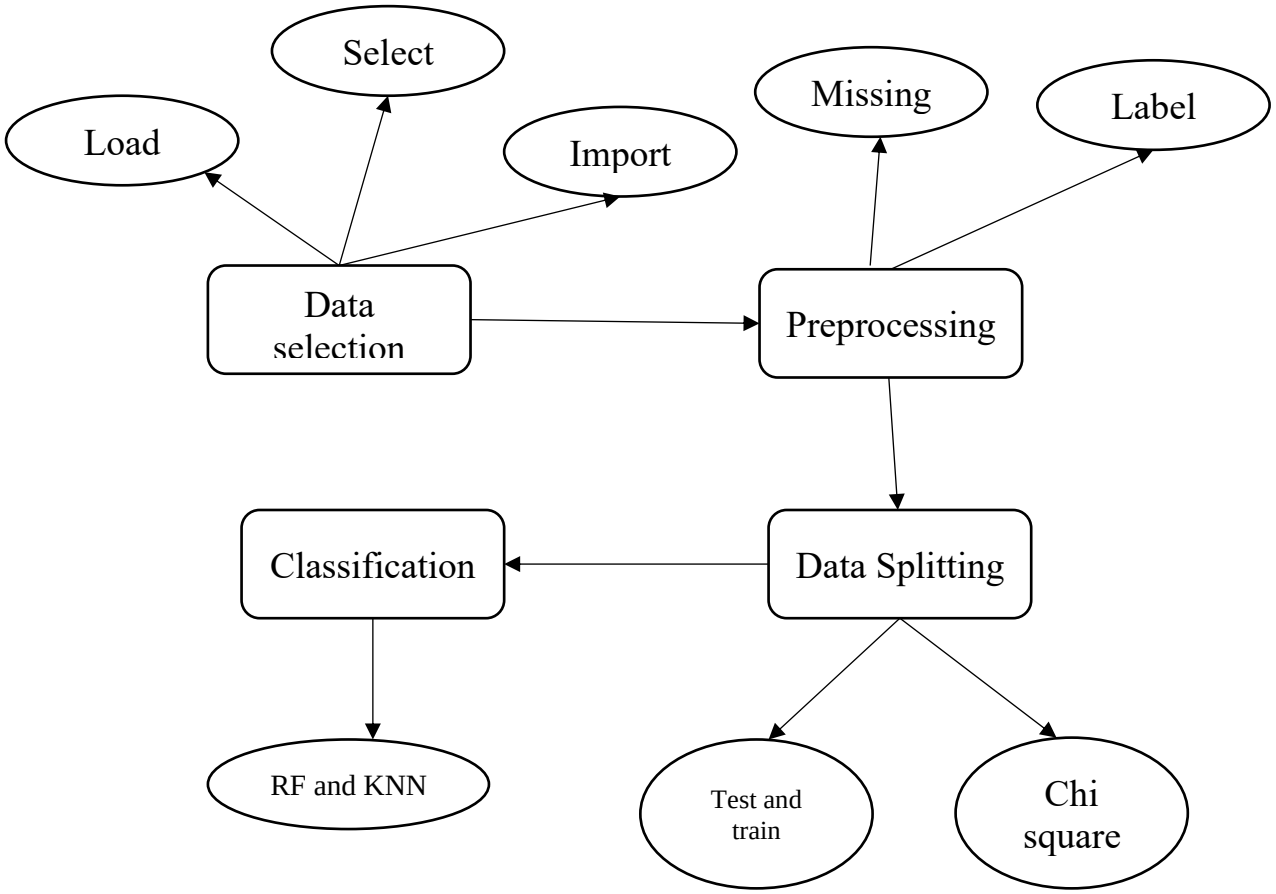


FIGURE 3.3.4: ER DIAGRAM

3.3.6 CLASS DIAGRAM

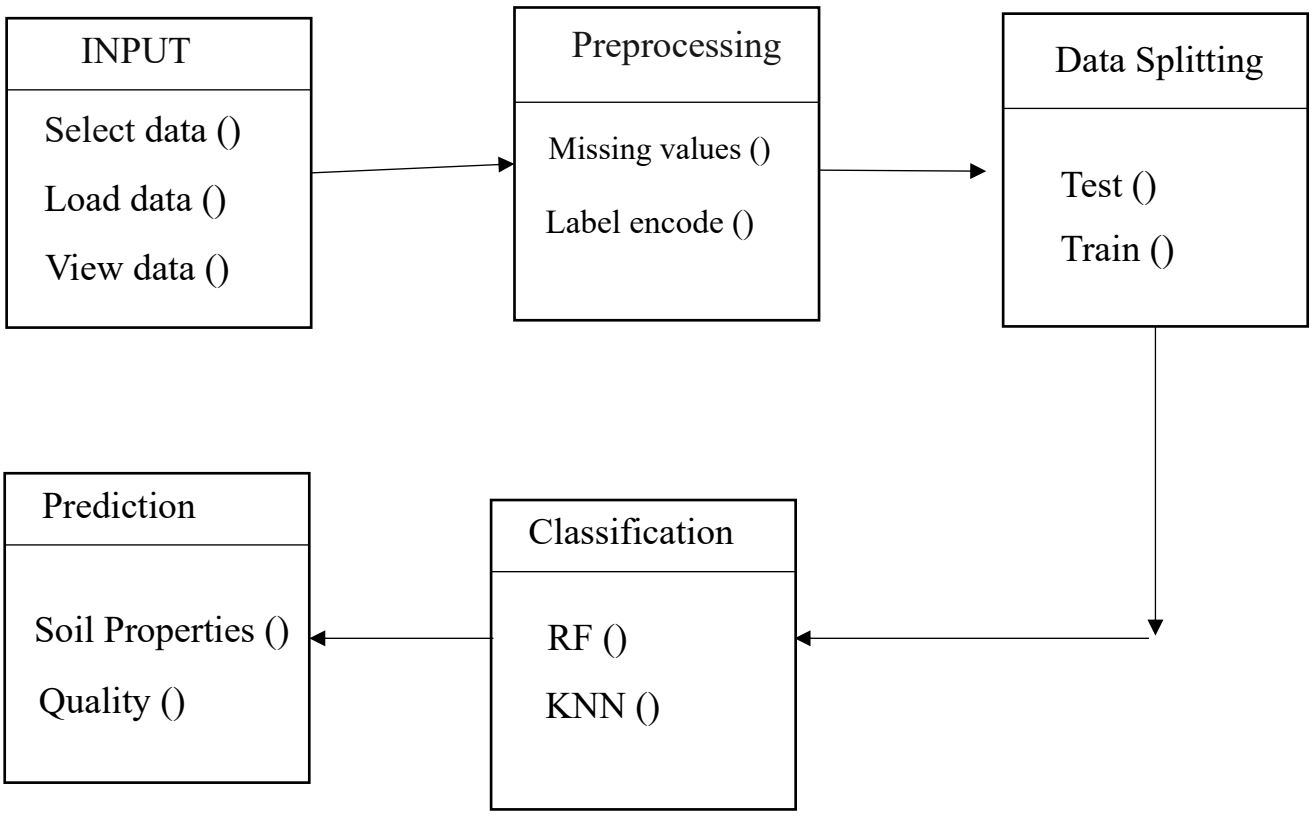


FIGURE 3.3.5: CLASS DIAGRAM

CHAPTER 4

IMPLEMENTATION

4.1 MODULES

- Data selection
- Data preprocessing
- Data splitting
- Feature selection
- Classification
- Result Generation

4.2 MODULES DESCRIPTION

4.2.1 DATA SELECTION

- The input data was collected from dataset repository.
- In this process, soil property dataset is used.
- The data selection is the process of predicting the soil properties.
- A collection of 1,886 soil sample measures is used for performance comparison of machine learning models. The soil was collected from a variety of locations in Africa. Each data point consists of 3,594 features:

1. PIDN: unique soil sample identifier
2. SOC: Soil organic carbon
3. pH: pH values
4. Ca: Mehlich-3 extractable Calcium
5. P: Mehlich-3 extractable Phosphorus
6. Sand: Sand content
7. m7497.96 - m599.76: There are 3,578 mid-infrared absorbance measurements. For example,”m599.76” column is the absorbance at wavenumber 599.76 cm-1.
8. Depth: Depth of the soil sample (2 categories: 1. ”Topsoil”, 2. ”Subsoil”)

9. BSA: Average long-term Black Sky Albedo measurements from MODIS satellite images (BSAN = nearinfrared, BSAS = shortwave, BSAV = visible)
10. CTI: Compound topographic index calculated from Shuttle Radar Topography Mission elevation data
11. ELEV: Shuttle Radar Topography Mission elevation data
12. EVI: Average long-term Enhanced Vegetation Index from MODIS satellite images

4.2.2 DATA PREPROCESSING

- Data pre-processing is the process of removing the unwanted data from the dataset.
- Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning.
- This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.
- Missing data removal
- Encoding Categorical data
- Missing data removal: In this process, the null values such as missing values and non values are replaced by 0.
- Missing and duplicate values were removed and data was cleaned of any abnormalities.
- Encoding Categorical data: That categorical data is defined as variables with a finite set of label values.
- That most machine learning algorithms require numerical input and output variables.

4.2.3 DATA SPLITTING

- During the machine learning process, data are needed so that learning can take place.
- In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how well it works.
- In this process, 70% of the dataset is to be the training data and the remaining 30% to be the testing data.
- Data splitting is the act of partitioning available data into two portions, usually for cross-validator purposes.
- One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.

4.2.5 CLASSIFICATION

- In this process, the machine learning algorithms such as random forest regression and KNN regression is implemented.
- **Random Forest Regression** is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.
- KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighborhood.

4.2.6 RESULT GENERATION

The Final Result will get generated based on the overall classification and prediction. The performance of this proposed approach is evaluated using some measures like,

- **MAE:** In statistics, the **mean absolute error** (MAE) is a way to measure the accuracy of a given model. It is calculated as:

Where:

$$\mathbf{MAE = (1/n) * \Sigma|y_i - x_i|}$$

  - **Σ:** A Greek symbol that means "sum"
  - $\mathbf{y_i}$**:** The observed value for the i^{th} observation
  - $\mathbf{x_i}$**:** The predicted value for the i^{th} observation
  - **n:** The total number of observations

- **MSE:** The mean squared error (MSE) is a common way to measure the prediction accuracy of a model. It is calculated as:

$$\mathbf{MSE} = (1/n) * \Sigma\,(\text{actual} - \text{prediction})^2$$

Where:

- **Σ** – a fancy symbol that means "sum"
- **n** – sample size
- **actual** – the actual data value
- **forecast** – the predicted data value

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

- System : Pentium IV 2.4 GHz
- Hard Disk : 200 GB
- Mouse : Logitech.
- Keyboard : 110 keys enhanced
- RAM : 4GB

5.2 SOFTWARE REQUIREMENTS

- O/S : Windows 7.
- Language : Python
- Front End : Anaconda Navigator – Spyder

5.3 SOFTWARE DESCRIPTION

5.3.1 Python

Python is one of those rare languages which can claim to be both *simple* and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. The official introduction to Python is Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

5.3.2 Features of Python

- **Simple**

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

- **Easy to Learn**

Python is extremely easy to get started with. Python has an extraordinarily simple syntax.

- **Free and Open Source**

Python is an example of a *FLOSS* (Free/Libré and Open Source Software). In simple terms, you can freely distribute copies of this software, read its source code, make changes to it, and use pieces of it in new free programs. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good - it has been created and is constantly improved by a community who just want to see a better Python.

- **High-level Language**

In Python, there is no need to bother about the low-level details such as managing the memory used by your program, etc.

- **Portable**

Due to its open-source nature, Python has been ported to (i.e. changed to make it work on) many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features.

Python can be used on GNU/Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and PocketPC!

Platform like Kivy can be used to create games for computer *and* for iPhone, iPad, and Android.

- **Interpreted**

A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by computer (binary code i.e. 0s and 1s) using a compiler with various flags and options. When run the program, the linker/loader software copies the program from hard disk to memory and starts running it.

Python, on the other hand, does not need compilation to binary. It just *run* the program directly from the source code. Internally, Python converts the source code into an intermediate form called bytecodes and then translates this into the native language of your computer and then runs it. All this, actually, makes using Python much easier since you don't have to worry about compiling the program, making sure that the proper libraries are linked and loaded, etc. This also makes Python programs much more portable; user can just copy Python program onto another computer and it just works!

- **Object Oriented**

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

- **Extensible**

If user need a critical piece of code to run very fast or want to have some piece of algorithm not to be open, they can code that part of your program in C or C++ and then use it from Python program.

- **Embeddable**

User can embed Python within your C/C++ programs to give scripting capabilities for program's users.

- **Extensive Libraries**

The Python Standard Library is huge indeed. It can help to do various things involving regular expressions, documentation generation, unit testing, threading, databases, web browsers, CGI, FTP, email, XML, XML-RPC, HTML, WAV files, cryptography, GUI (graphical user interfaces), and other system-dependent stuff. Remember, all this is always available wherever Python is installed. This is called the Batteries Included philosophy of Python.

Besides the standard library, there are various other high-quality libraries which you can find at the Python Package Index.

5.4 TESTING PRODUCTS

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. . A series of tests are performed before the system is ready for the user

acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that "al gears mesh", that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.

5.4.1 UNIT TESTING

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

5.4.2 INTEGRATION TESTING

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. Need for the integrated test is to find the overall system performance. There are two types of integration testing. They are:

i) Top-down integration testing.

ii) Bottom-up integration testing.

5.4.3 TESTING TECHNIQUES/STRATEGIES

- **WHITE BOX TESTING**

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing meth Derived test cases that guarantee that all independent paths within a module have been exercised at least once.

- **BLACK BOX TESTING**

1. Black box testing is done to find incorrect or missing function
2. Interface error
3. Errors in external database access
4. Performance errors.
5. Initialization and termination errors

In 'functional testing', it is performed to validate an application conforms to its specifications of correctly performs all its required functions. So, this testing is also called 'black box testing'. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

5.4.4 SOFTWARE TESTING STRATEGIES

VALIDATION TESTING

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, But a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

USER ACCEPTANCE TESTING

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

OUTPUT TESTING

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

CHAPTER 6
CONCLUSION

The soil property dataset is taken as input. It is used for pre-processing step to avoid wrong prediction. This work also implement the machine learning algorithm such as KNN regression and random forest regression for better performance. This is a time series dataset to implement the machine learning algorithm. The experimental results shows that the performance metrics such as MAE, MSE has high accuracy.

CHAPTER 7
FUTURE ENHANCEMENT

In future, this work will be extended to hybrid the two different machine learning, and also provide extensions or modifications to the proposed clustering and classification algorithms to achieve further increased performance. Apart from the experimented combination of data mining techniques, further combinations and other clustering algorithms can be used to improve the detection accuracy.

CHAPTER 8
SAMPLE CODING

```
#=====IMPORT PACKAGES ============================
import pandas as pd
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
#======================== DATA SELECTION
=========================
print("=========================================")
print("---------- Data Selection ------------")
print("=========================================")
data=pd.read_csv('training.csv')
print(data.head(10))

print()

data=data[['Depth','Ca','P','pH','SOC','Sand']]
#==================== PREPROCESSING
======================================
#checking missing values
print("=========================================")
print("--------- Checking missing values ------------")
print("=========================================")
print(data.isna().sum())
print()
#========DATA SPLITTING ============================
x=data.drop('Sand',axis=1)
```

```
y=data['Sand']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
random_state=2)
print("------------------------------------------------------------")
print("======================== Data splitting
====================")
print("------------------------------------------------------------")
print()
print("Total No of input data    :",data.shape[0])
print()
print("Total No of training data :",X_train.shape[0])
print()
print("Total No of testing data  :",X_test.shape[0])
print()
#============== CLASSIFICATION ====================
from sklearn.neighbors import KNeighborsRegressor
from sklearn import metrics
#=== KNN regression ===
#initialize the model
knn = KNeighborsRegressor()

#fitting the model

knn.fit(X_train, y_train)

#predict the model

y_pred = knn.predict(X_test)

print("------------------------------------------------------------")
print("======================== KNN REGRESSION======")
print("------------------------------------------------------------")
```

```
print()
mae_ridge=metrics.mean_absolute_error(y_test, y_pred)
mse_ridge=metrics.mean_squared_error(y_test, y_pred)
print()
print("1.Mean Squared Error  : ",mse_ridge)
print()
print("1.Mean Absolute Error : ",mae_ridge)
#===== Random Forest Regression Model =======
from sklearn.ensemble import RandomForestRegressor
  # create regressor object
regressor = RandomForestRegressor(n_estimators = 100, random_state
= 0)
 # fit the regressor with x and y data
regressor.fit(X_train, y_train)
y_pred_rf=regressor.predict(X_test)
print("------------------------------------------------------------")
print("============== RANDOM FOREST REGRESSION ====")
print("------------------------------------------------------------")
print()
mae_rf=metrics.mean_absolute_error(y_pred_rf,y_test)
print("1.Mean Absolute Error :",mae_rf)
print()
mse_rf=metrics.mean_squared_error(y_pred_rf,y_test)
x=data.drop('Sand',axis=1)
y=data['Sand']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
random_state=2)

print("------------------------------------------------------------")
```

```
print("======================== Data splitting ==========")
print("-----------------------------------------------------")
print()
print("Total No of input data    :",data.shape[0])
print()
print("Total No of training data :",X_train.shape[0])
print()
print("Total No of testing data  :",X_test.shape[0])
print()
#================= CLASSIFICATION ================
from sklearn.neighbors import KNeighborsRegressor
from sklearn import metrics
#=== ridge regression ===
#initialize the model
knn = KNeighborsRegressor()
#fitting the model
knn.fit(X_train, y_train)
from sklearn.ensemble import RandomForestRegressor
 # create regressor object
regressor = RandomForestRegressor(n_estimators = 100, random_state
= 0)
# fit the regressor with x and y data
regressor.fit(X_train, y_train)
y_pred_rf=regressor.predict(X_test)
print("-----------------------------------------------------")
print("============== RANDOM FOREST REGRESSION
====================")
print("-----------------------------------------------------")
print()
```

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn import metrics
#=== KNN regression ===
#initialize the model
knn = KNeighborsRegressor()

#fitting the model

knn.fit(X_train, y_train)

#predict the model

y_pred = knn.predict(X_test)

print("-------------------------------------------------------")
print("============== KNN REGRESSION ==============")
print("-------------------------------------------------------")
print()
mae_rf=metrics.mean_absolute_error(y_pred_rf,y_test)
print("1.Mean Absolute Error :",mae_rf)
print()
mse_rf=metrics.mean_squared_error(y_pred_rf,y_test)
print("2.Mean Squared Error :",mse_rf)
print()
#predict the model
y_pred = knn.predict(X_test)
print("2.Mean Squared Error :",mse_rf)
print()
print("-------------------------------------------------------")
print("======================== PREDICTION
========================")
```

```
print("-----------------------------------------------------")
print()
for i in range(0,10):
    Results=y[i]
    print()
    print([i],"The soil Properties is ", Results)
    print()
#========= PREDICTION ============================
import numpy as np
print()
print("-------------------------------------------------------")
print()
print("======== Input data 1 =============")
print()
input_1 = np.array([0,-0.295749,-0.0413364,-
1.12937,0.353258]).reshape(1, -1)
print()
print("The Actuall input data is : ",input_1)
predicted_data = knn.predict(input_1)
print()
print("The prediction is : ", predicted_data)
print()
 print("============================================")
print("------------------ Analyse the Crop  -------------")
print("============================================")
print()
mini=min(y_pred)
maxx=max(y_pred)
avg=(mini + maxx)/2
```

```
for i in range(0,10):
    if y_pred[i]<=mini:
        # print("============================")
        print()
        print([i],'The Soil Quality is LOW(POOR)')
        print()
        print("============================")
    else:
        # print("============================")
        print()
        print([i],'Soil Quality is GOOD ')
        print()
        print("============================")
import matplotlib.pyplot as plt
vals=[mae_ridge,mae_rf]
inds=range(len(vals))
labels=["KNN","Random forest"]
fig,ax = plt.subplots()
rects = ax.bar(inds, vals)
ax.set_xticks([ind for ind in inds])
ax.set_xticklabels(labels)
plt.show()
```

SAMPLE SCREENSHOTS

```
====================================
---------- Data Selection ------------
====================================
      PIDN  m7497.96  m7496.04  ...        pH       SOC      Sand
0  XNhoFZW5  0.302553  0.301137  ... -1.129366  0.353258  1.269748
1  9XNspFTd  0.270192  0.268555  ... -1.531538 -0.264023  1.692209
2  WDId41qG  0.317433  0.316265  ... -0.259551  0.064152  2.091835
3  JrrJf1mN  0.261116  0.259767  ... -0.577548 -0.318719  2.118477
4  ZoIitegA  0.260038  0.258425  ... -0.699135 -0.310905  2.164148
5  QkhF9azr  0.172350  0.170766  ... -0.914251 -0.685962  1.943402
6  kFpRoFpt  0.354537  0.352801  ... -0.839428  0.064152  1.441016
7  QH0y7Gc0  0.312328  0.310354  ... -1.157424 -0.217141  0.866316
8  yTfzYmjL  0.351604  0.349781  ...  0.021034  0.634551  0.645571
9  9my11nPJ  0.321137  0.319281  ... -0.147317 -0.209327  1.753105

[10 rows x 3600 columns]
```

```
============================================
--------- Checking missing values ----------
============================================
Depth    0
Ca       0
P        0
pH       0
SOC      0
Sand     0
dtype: int64
```

```
------------------------------------------------------------
====================== Data splitting ======================
------------------------------------------------------------

Total No of input data    : 1157

Total No of training data : 809

Total No of testing data  : 348
```

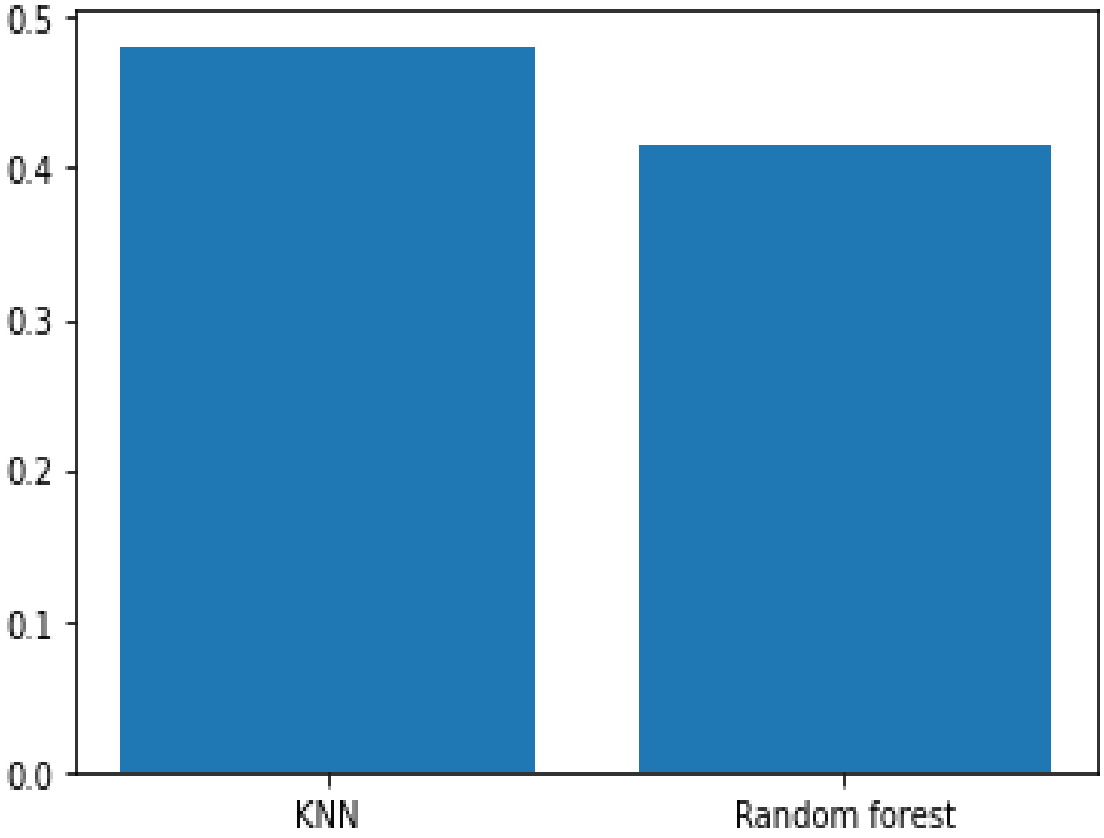
```
------------------------------------------------------------

====================== KNN REGRESSION ======================

------------------------------------------------------------



1.Mean Squared Error  :  0.4354236377667784

1.Mean Absolute Error :  0.48061821146273437
------------------------------------------------------------
```

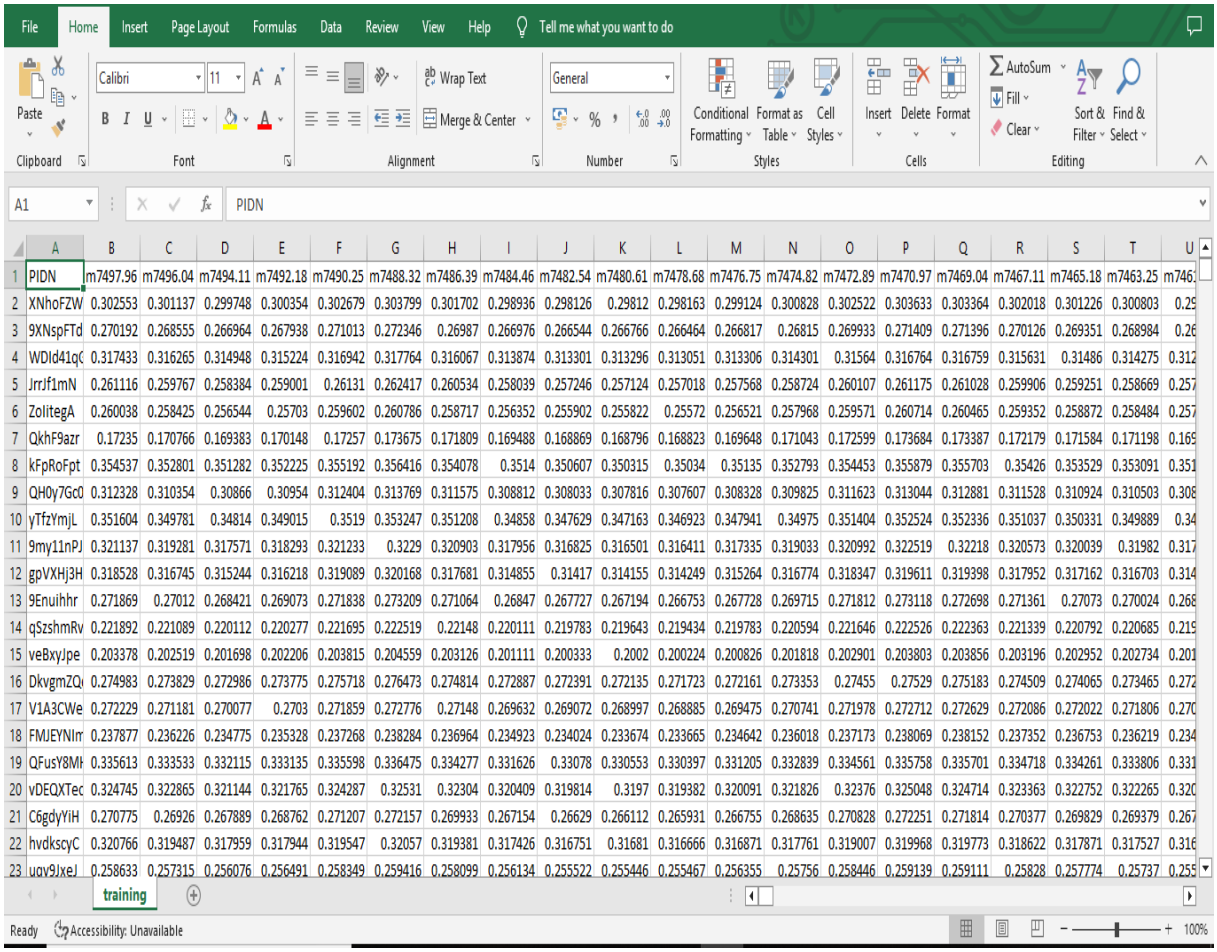
```
------------------------------------------------------------
============== RANDOM FOREST REGRESSION ====================
------------------------------------------------------------

1.Mean Absolute Error : 0.4138623524866611

2.Mean Squared Error : 0.3170729311255068
```

```
------------------------------------------------------------
======================== PREDICTION ========================
------------------------------------------------------------


[0] The soil Properties is  1.26974781214839

[1] The soil Properties is  1.69220922517683

[2] The soil Properties is  2.09183488614967

[3] The soil Properties is  2.1184765968812003

[4] The soil Properties is  2.1641481009923798

[5] The soil Properties is  1.94340249778833

[6] The soil Properties is  1.4410159525653201

[7] The soil Properties is  0.8663161924996129
```

```
------------------------------------------------------------
============== RANDOM FOREST REGRESSION ====================
------------------------------------------------------------

1.Mean Absolute Error : 0.4138623524866611

2.Mean Squared Error : 0.3170729311255068
```

```
[0] Soil Quality is GOOD

============================

[1] Soil Quality is GOOD

============================

[2] Soil Quality is GOOD

============================

[3] Soil Quality is GOOD

============================

[4] Soil Quality is GOOD

============================

[5] Soil Quality is GOOD

============================

[6] Soil Quality is GOOD

----------------------------
```


0.5
0.4
0.3
0.2
0.1
0.0
KNN
Random forest



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	PIDN	m7497.96	m7496.04	m7494.11	m7492.18	m7490.25	m7488.32	m7486.39	m7484.46	m7482.54	m7480.61	m7478.68	m7476.75	m7474.82	m7472.89	m7470.97	m7469.04	m7467.11	m7465.18	m7463.25	m746
2	XNhoFZW	0.302553	0.301137	0.299748	0.300354	0.302679	0.303799	0.301702	0.298936	0.298126	0.29812	0.298163	0.299124	0.300828	0.302522	0.303633	0.303364	0.302018	0.301226	0.300803	0.29
3	9XNspFTd	0.270192	0.268555	0.266964	0.267938	0.271013	0.272346	0.26987	0.266976	0.266544	0.266766	0.266464	0.266817	0.26815	0.269933	0.271409	0.271396	0.270126	0.269351	0.268984	0.26
4	WDId41qC	0.317433	0.316265	0.314948	0.315224	0.316942	0.317764	0.316067	0.313874	0.313301	0.313296	0.313051	0.313306	0.314301	0.31564	0.316764	0.316759	0.315631	0.31486	0.314275	0.312
5	JrrJf1mN	0.261116	0.259767	0.258384	0.259001	0.26131	0.262417	0.260534	0.258039	0.257246	0.257124	0.257018	0.257568	0.258724	0.260107	0.261175	0.261028	0.259906	0.259251	0.258669	0.257
6	ZolitegA	0.260038	0.258425	0.256544	0.25703	0.259602	0.260786	0.258717	0.256352	0.255902	0.255822	0.25572	0.256521	0.257968	0.259571	0.260714	0.260465	0.259352	0.258872	0.258484	0.257
7	QkhF9azr	0.17235	0.170766	0.169383	0.170148	0.17257	0.173675	0.171809	0.169488	0.168869	0.168796	0.168823	0.169648	0.171043	0.172599	0.173684	0.173387	0.172179	0.171584	0.171198	0.169
8	kFpRoFpt	0.354537	0.352801	0.351282	0.352225	0.355192	0.356416	0.354078	0.3514	0.350607	0.350315	0.35034	0.35135	0.352793	0.354453	0.355879	0.355703	0.35426	0.353529	0.353091	0.351
9	QH0y7Gc0	0.312328	0.310354	0.30866	0.30954	0.312404	0.313769	0.311575	0.308812	0.308033	0.307816	0.307607	0.308328	0.309825	0.311623	0.313044	0.312881	0.311528	0.310924	0.310503	0.308
10	yTfzYmjL	0.351604	0.349781	0.34814	0.349015	0.3519	0.353247	0.351208	0.34858	0.347629	0.347163	0.346923	0.347941	0.34975	0.351404	0.352524	0.352336	0.351037	0.350331	0.349889	0.34
11	9my11nPJ	0.321137	0.319281	0.317571	0.318293	0.321233	0.3229	0.320903	0.317956	0.316825	0.316501	0.316411	0.317335	0.319033	0.320992	0.322519	0.32218	0.320573	0.320039	0.31982	0.317
12	gpVXHj3H	0.318528	0.316745	0.315244	0.316218	0.319089	0.320168	0.317681	0.314855	0.31417	0.314155	0.314249	0.315264	0.316774	0.318347	0.319611	0.319398	0.317952	0.317162	0.316703	0.314
13	9Enuihhr	0.271869	0.27012	0.268421	0.269073	0.271838	0.273209	0.271064	0.26847	0.267727	0.267194	0.266753	0.267728	0.269715	0.271812	0.273118	0.272698	0.271361	0.27073	0.270024	0.268
14	qSzshmRv	0.221892	0.221089	0.220112	0.220277	0.221695	0.222519	0.22148	0.220111	0.219783	0.219643	0.219434	0.219783	0.220594	0.221646	0.222526	0.222363	0.221339	0.220792	0.220685	0.219
15	veBxyJpe	0.203378	0.202519	0.201698	0.202206	0.203815	0.204559	0.203126	0.201111	0.200333	0.2002	0.200224	0.200826	0.201818	0.202901	0.203803	0.203856	0.203196	0.202952	0.202734	0.201
16	DkvgmZQ	0.274983	0.273829	0.272986	0.273775	0.275718	0.276473	0.274814	0.272887	0.272391	0.272135	0.271723	0.272161	0.273353	0.27455	0.27529	0.275183	0.274509	0.274065	0.273465	0.272
17	V1A3CWe	0.272229	0.271181	0.270077	0.2703	0.271859	0.272776	0.27148	0.269632	0.269072	0.268997	0.268885	0.269475	0.270741	0.271978	0.272712	0.272629	0.272086	0.272022	0.271806	0.270
18	FMJEYNIm	0.237877	0.236226	0.234775	0.235328	0.237268	0.238284	0.236964	0.234923	0.234024	0.233674	0.233665	0.234642	0.236018	0.237173	0.238069	0.238152	0.237352	0.236753	0.236219	0.234
19	QFusY8Mk	0.335613	0.333533	0.332115	0.333135	0.335598	0.336475	0.334277	0.331626	0.33078	0.330553	0.330397	0.331205	0.332839	0.334561	0.335758	0.335701	0.334718	0.334261	0.333806	0.331
20	vDEQXTec	0.324745	0.322865	0.321144	0.321765	0.324287	0.32531	0.32304	0.320409	0.319814	0.3197	0.319382	0.320091	0.321826	0.32376	0.325048	0.324714	0.323363	0.322752	0.322265	0.320
21	C6gdyYiH	0.270775	0.26926	0.267889	0.268762	0.271207	0.272157	0.269933	0.267154	0.26629	0.266112	0.265931	0.266755	0.268635	0.270828	0.272251	0.271814	0.270377	0.269829	0.269379	0.267
22	hvdkscyC	0.320766	0.319487	0.317959	0.317944	0.319547	0.32057	0.319381	0.317426	0.316751	0.31681	0.316666	0.316871	0.317761	0.319007	0.319968	0.319773	0.318622	0.317871	0.317527	0.316
23	uqv9JxeJ	0.258633	0.257315	0.256076	0.256491	0.258349	0.259416	0.258099	0.256134	0.255522	0.255446	0.255467	0.256355	0.25756	0.258446	0.259139	0.259111	0.25828	0.257774	0.25737	0.255

CHAPTER 10

REFERENCES

[1] J. v. B. Ephraim Nkonya, Alisher Mirzabaev, Economics of Land Degradation and Improvement – A Global Assessment for Sustainable Development, Spinger, 2016.

[2] B. S. B. Arun Kumar, A. Kumar, "Prospective of Indian agriculture: highly vulnerable to huge unproductivity and unsustainability, current science vol. 119, no. 7, pp. 1079–1080, 2020.

[3] I. Barra, S. Haefele, R. Sakrabani, F. Kebede, "Soil spectroscopy with the use of chemometrics, machine learning and pre-processing techniques in soil diagnosis: Recent advances -a review", TrAC Trends in Analytical Chemistry, vol. 135, 2020.

[4] B. Patra, R. Pal, P. Rajamani, P. Surya, "Mineralogical composition and c/n contents in soil and water among betel vineyards of coastal odisha, india", SN Applied Sciences, vol. 2, 2020.

[5] I. Barra, S. M. Haefele, R. Sakrabani, F. Kebede, "Soil spectroscopy with the use of chemometrics, machine learning and pre-processing techniques in soil diagnosis: Recent advances–a review", TrAC Trends in Analytical Chemistry, vol. 135, p. 116166, 2021.

[6] H. Yu, D. Liu, G. Chen, B. Wan, S. Wang, B. Yang, "A neural network ensemble method for precision fertilization modeling", Mathematical and Computer Modelling, vol. 51, no. 11, pp. 1375–1382, 2010.

[7] I. Sydorenko, "Training data", https://labelyourdata.com/ articles/machine-learning-and-training-data, 2021.

[8] S. Ray, “A quick review of machine learning algorithms”, “2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)”, pp. 35–39, 2019,

[9] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. Mclachlan, A. Ng, B. Liu, P. S. Yu, et al., “Top 10 algorithms in data mining”, Knowledge and Information Systems, vol. 14, no. 1, p. 1–37, 2007.

[10] T. K. Ho, “Random decision forests”, “Proceedings of 3rd International Conference on Document Analysis and Recognition”, vol. 1, pp. 278–282 vol.1, 1995.

[11] T. K. Ho, “The random subspace method for constructing decision forests”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832–844, 1998.

[12] T. Hastie, J. Friedman, R. Tisbshirani, The Elements of statistical learning: data mining, inference, and prediction, Springer, 2017.

[13] S. M. Piryonesi, T. E. El-Diraby, “Data analytics in asset management: Cost-effective prediction of the pavement condition index”, Journal of Infrastructure Systems, vol. 26, no. 1, p. 04019036, 2020.

[14] C. Cortes, V. Vapnik, “Support-vector networks”, Machine Learning, vol. 20, no. 3, p. 273–297, 1995.