SOIL PROPERTIES PREDICTION BY USING MACHINE LEARNING

BY

Jasmin Babariya

(Student id: 202319001)

Darshan Gajera

(Student id: 202319006)

Harsh Chothani

(Student id: 202319010)

Deep Pethani

(Student id: 202319011)

Smit Bhuva

(Student id: 202319027)

MASTER OF SCIENCE IN AGRICULTURE ANALYTICS

SUBMITTED TO

Dr. Ranendu Ghosh

Dhirubhai Ambani Institute of Information

and Communication Technology, Gujarat.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

Acronym	Full Form
N	Nitrogen
P	Phosphorus
K	Potassium
Ca	Calcium
Mg	Magnesium
S	Sulphur
Fe	Iron
Mn	Manganese
Zn	Zinc
Cu	Copper
B	Boron
Mo	Molybdenum
Cl	Chlorine
CEC	Cation Exchange Capacity
EC	Electrical Conductivity
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
SOC	Soil Organic Carbon
GBDT	Gradient Boosting Decision Tree
NDVI	Normalized Difference Vegetation Index
DEM	Digital Elevation Model
LST	Land Surface Temperature
GIS	Geographic Information System
QGIS	Quantum Geographic Information System
CSV	Comma Separated Value
SHP	Shape File
TIFF	Tagged Image File Format

ABSTRACT

This study presents a comprehensive exploration of soil nutrient mapping through the integration of Google Earth Engine data and advanced modeling techniques. Leveraging the Scorpan model, supervised classification, and linear regression, our research focuses on predicting soil nutrient values using key features-Normalized Difference Vegetation Index (NDVI), Digital Elevation Model (DEM), and Land Surface Temperature (LST). The NDVI, DEM, and LST files crucial for this analysis are sourced from Google Earth Engine, emphasizing the accessibility and utility of remote sensing data.

The workflow begins with the application of supervised classification algorithms to categorize soil samples based on Scorpan model-derived features. Subsequently, a linear regression model is employed to establish quantitative relationships between NDVI, DEM, LST, and soil nutrient levels. This tailored approach aims to generate diverse soil nutrient maps that cater to the unique characteristics of different geographical regions.

By utilizing Google Earth Engine data, our study not only streamlines the acquisition of essential input features but also showcases the potential for integrating freely available satellite data into sophisticated modeling frameworks. This approach enhances the accuracy and specificity of soil nutrient predictions, contributing to a more nuanced understanding of soil composition and environmental conditions.

1. INTRODUCTION

1.1 Soil Nutrient

Soil nutrients refer to the essential elements and compounds present in the soil that are necessary for the growth and development of plants. These nutrients play a crucial role in supporting various physiological processes in plants, ultimately influencing their overall health and productivity. The key soil nutrients can be broadly categorized into two groups: macronutrients and micronutrients.

- **Macronutrients**

Plants need special nutrients, called macronutrients, in bigger amounts to grow well. Nitrogen(N) helps plants make proteins and grow big, Phosphorus(P) gives them energy and helps roots and flowers, and Potassium(K) keeps plants strong and helps them take in water for making food. Calcium(Ca) supports the structure of plant cells, and Magnesium(Mg) helps in making leaves green. Sulphur(S) is important for making proteins. All these nutrients work together to make sure plants grow strong and healthy, giving us good crops.

- **Micronutrients**

These are required in smaller quantities but are equally vital for plant development. Examples of micronutrients include iron (Fe), manganese (Mn), zinc (Zn), copper (Cu), boron (B), molybdenum (Mo), and chlorine (Cl). Each micronutrient plays a specific role in various biochemical and physiological processes within the plant.

The availability and balance of these nutrients in the soil directly impact the health and yield of crops. Soil nutrient analysis is crucial for farmers and land managers to assess the nutrient content of the soil and make informed decisions about fertilization and soil management practices to optimize plant growth and productivity.

1.2 Soil Chemical Properties

- **PH:** Measures the acidity or alkalinity of the soil.
- **Nutrient Content:** Includes concentrations of essential elements such as nitrogen (N), phosphorus (P), potassium (K), calcium (Ca), magnesium (Mg), sulphur (S), and micronutrients (iron, manganese, zinc, copper, boron, molybdenum, chlorine).
- **Cation Exchange Capacity (CEC):** Reflects the soil's ability to hold and exchange cations (positively charged ions).
- **Electrical Conductivity (EC):** Measures the soil's ability to conduct an electrical current, indicating salinity levels.
- **Organic Matter Content:** The percentage of decomposed plant and animal material in the soil.
- **Soil Colour:** It Can provide clues about the presence of certain minerals and organic matter.

1.3 Objectives

➢ To create a smart computer program using machine learning that can tell us about the nutrients in the soil by using SCROPAN model. This will help farmers know how to take care of their land better for growing crops.

➢ To empower farmers to make decisions for sustainable crop production and soil management, we will leverage machine learning to enhance soil nutrient prediction.

➢ To significantly impact agriculture and related industries, the project focuses on accurately predicting soil nutrient levels. This precision in forecasting will enhance the effectiveness of crop nutrient management strategies, providing valuable insights for advancing agricultural practices and ultimately boosting crop productivity.

2. REVIEW OF LITERATURE

2.1 Digital mapping to extrapolate the selected soil fertility attributes

Pegah Khosravani, *et al* (2023) researched on Digital mapping to extrapolate the selected soil fertility attributes in calcareous soils of a semiarid region in Iran. Spatial variability of soil properties is considered as one of the most important reasons for the variability of crop productions. The current research was conducted to determine the capability of machine learning models for the generalization of the modeling results from the reference area, i.e., Marvdasht plain, for estimating soil fertility attributes with the aims of extrapolating the modeling results to receptor area, i.e., Bandamir, in Iran with the aid of Homosoil concepts.

2.2 Mapping Soil Organic Carbon in Floodplain Farmland

Dr. Zihao Wu, *et.al* (2023) studied on Mapping Soil Organic Carbon in Floodplain Farmland. Accurately mapping soil organic carbon (SOC) is conducive to evaluating carbon storage and soil quality. However, the high spatial heterogeneity of SOC caused by river-related factors and agricultural management brings challenges to digital soil mapping in floodplain farmland. Moreover, current studies focus on the non-linear relationship between SOC and covariates, but ignore the effective range of environmental variables on SOC, which prevents the revelation of the SOC differentiation mechanism. Using the 375 samples collected from the Jiangchang Town near Han River, we aim to determine the main controlling factors of SOC, reveal the effective range of environmental variables, and obtain the spatial map of SOC by using the gradient boosting decision tree (GBDT) model and partial dependence plots. Linear regression was used as a reference. Results showed that GBDT outperformed linear regression. GBDT results show that the distance from the river was the most important SOC factor, confirming the importance of the Han River to the SOC pattern. The partial dependence plots indicate that all environmental variables have their effective ranges, and when their values are extremely high or low, they do not respond to changes in SOC. Specifically, the influential ranges of rivers, irrigation canals, and rural settlements on SOC were within 4000, 200, and 50 m, respectively. The peak SOC was obtained with high clay (≥31%), total nitrogen (≥1.18 g/kg), and total potassium contents (≥11.1 g/kg), but it remained steady when these covariates further increased. These results highlight the importance of revealing the effective range of environmental variables, which provides data support for understanding the spatial pattern of

SOC in floodplain farmland, achieving carbon sequestration in farmland and precision agriculture. The GBDT with the partial dependence plot was effective in SOC fitting and mapping.

3. MATERIALS & METHODS

3.1 Scorpan Model

The scorpan model is a widely used soil classification system that divides the Earth's land into distinct areas based on their geological, biological, and environmental characteristics. It is particularly useful for soil nutrient mapping as it provides a standardized framework for categorizing soils and understanding their nutrient levels.

- **Soil categorization**

The scorpan model helps in categorizing soil samples into different classes based on their physical, chemical, and biological properties. This categorization is essential for understanding the nutrient levels and environmental conditions of the soil.

- **Nutrient estimation**

The scorpan model can be used to estimate soil nutrient levels by analyzing the chemical properties of soil samples and comparing them with the reference soil data. This estimation is crucial for determining the optimal nutrient management strategies for farmers and land managers.

- **Soil-to-vegetable mapping**

By combining the scorpan model with remote sensing data, such as Normalized Difference Vegetation Index (NDVI), Digital Elevation Model (DEM), and Land Surface Temperature (LST), the model can be used to map soil nutrient levels across a region. This mapping helps in identifying areas with high nutrient levels, which can be targeted for sustainable agriculture practices.

- **Integration with machine learning**

The scorpan model can be used in conjunction with machine learning algorithms for soil nutrient mapping. By training the algorithms on the Scorpan model's soil classes and nutrient levels, more accurate and detailed soil nutrient maps can be generated.

The scorpan model is a valuable tool for soil nutrient mapping as it provides a standardized framework for categorizing soils and estimating their nutrient levels. By integrating the scorpan model with remote sensing data and machine learning techniques, more accurate and detailed soil nutrient maps can be generated, which can be used to inform sustainable agriculture practices and environmental conservation strategies.

3.2 Study Area

In this project, study area of Vadodara District, extending across a vast expanse of 7,776 square kilometers (or 777,600 hectares). Positioned at coordinates 73.19° E longitude and 22.30° N latitude, the district showcases a diverse landscape within Gujarat, India. The study area varies in altitude, ranging from approximately 30 to 160 meters above mean sea level.

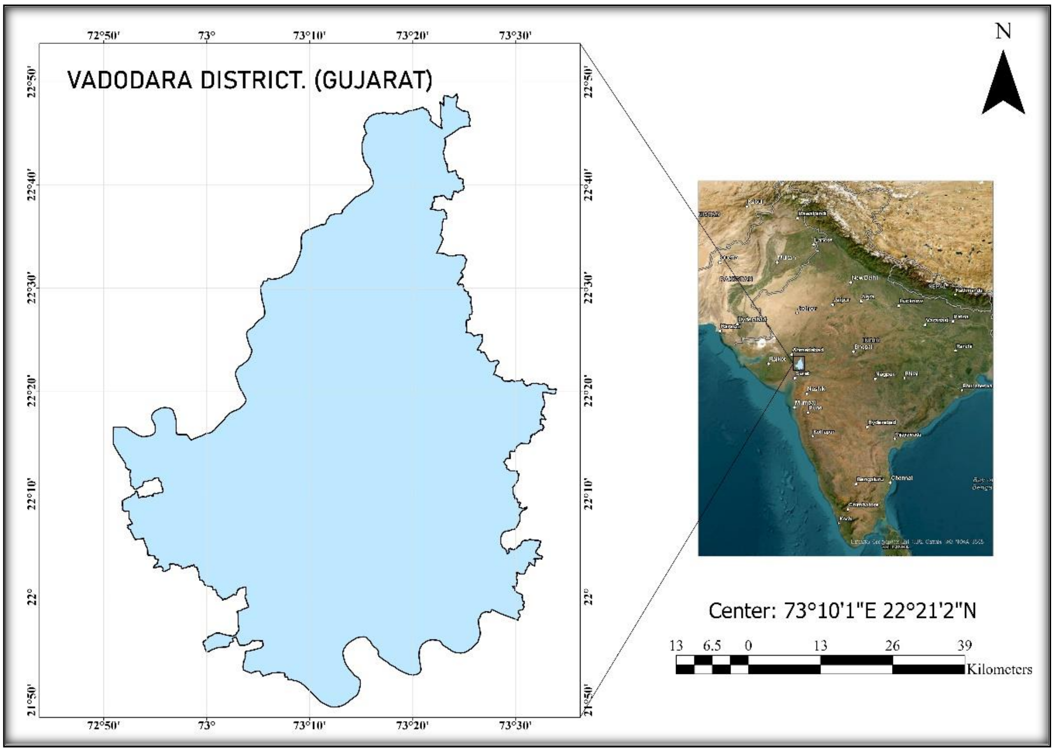

FIG : 1 Vadodara District Study Area Image

3.3 Datasets

1. NDVI

NDVI, or Normalized Difference Vegetation Index, is a widely used remote sensing technique that measures the health and vitality of vegetation. It is calculated from satellite or aerial imagery by taking the normalized difference between near-infrared and red reflectance values. NDVI values range from -1 to 1, with higher values indicating denser and healthier vegetation cover.

Using satellite data from Google Earth to create a special map called NDVI, which shows how healthy plants are. It looks at a place called Vadodara in India from January to

December in 2017. The map uses different colors to show where plants are really healthy (like dark green) and where they might need more care (like light green).

2. DEM

DEM stands for Digital Elevation Model. It's like a digital map that shows how high or low the land is in a specific area. DEMs help us understand the shape and elevation of the Earth's surface, making them useful for various applications like mapping, planning, and environmental analysis.

DEM which is derived from Landsat dataset from google earth engine by a script code of months are from date 01/01/2017 to 31/12/2017.

3. LST

LST, or Land Surface Temperature, is a measure of how hot or cold the Earth's surface is. It provides valuable insights for understanding temperature variations across landscapes and is often used in environmental studies and analysis.

LST which is derived from Landsat dataset from google earth engine by a script code of months are from date 01/01/2017 to 31/12/2017.

4. CSV File

This CSV file captures a comprehensive dataset for Vadodara district, comprising 321 data points. It includes essential information such as taluka, village, latitude, and longitude coordinates. Additionally, the file provides key soil parameters such as electrical conductivity , pH levels, and concentrations of essential elements like Sulphur, Zinc, Copper, Boron, among others.

5. Vadodara District SHP File

To delineate Vadodara district from the India village boundary shapefile in ArcGIS Pro, the process involves utilizing the Dissolve tool. After downloading the India village boundary shapefile, it is imported into ArcGIS Pro, and Vadodara district is selected either manually or through attribute-based selection. Subsequently, the Dissolve tool is applied, specifying a unique identifier such as district name or code. This operation consolidates the selected features, effectively creating a single polygon that represents Vadodara district within the larger dataset. The resulting shapefile or feature class provides a simplified and distinct representation of Vadodara district boundaries for further analysis or mapping purposes.

➢ The complete set of required code is available in the appendix for reference.

3.4 Methodology

❖ **Normalized the images**

- ➢ Importing the NDVI GeoTIFF file into QGIS.
- ➢ Open the raster calculator tool.
- ➢ Apply the following normalization formula to achieve values between 0 and 1.
- ➢ Expression = (NDVI - NDVI_min) / (NDVI_max - NDVI_min)
- ➢ Execute the raster calculator to perform the normalization. Save the result as a new GeoTIFF file.
- ➢ Same also normalized the DEM and LST tiff files.

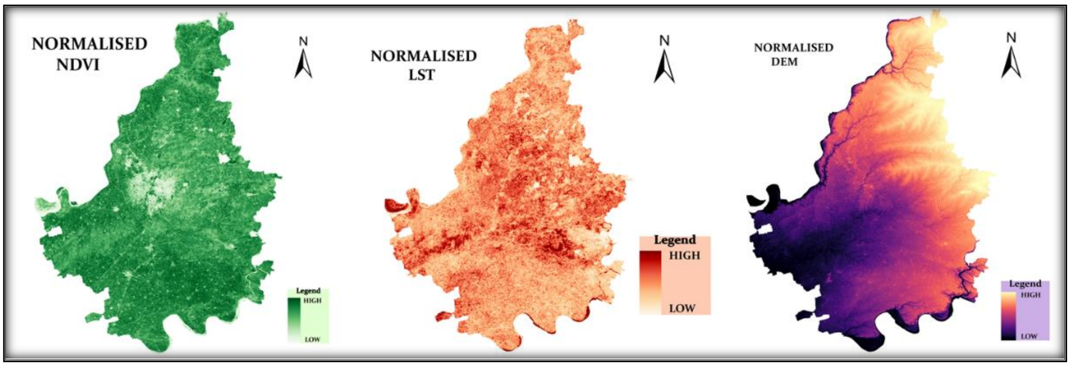

FIG : 2 Normalized Images (NDVI,DEM,LST)

❖ **Generate a sample point from the csv file**

To visualize sample data for Vadodara district in ArcGIS Pro, a simple process involves uploading a CSV file containing latitude and longitude information. The CSV file is imported into ArcGIS Pro, and the "Display XY Data" tool is utilized. This tool assigns the specified longitude and latitude columns from the CSV file to X and Y coordinates, respectively, generating points on the map corresponding to the geographical locations in the dataset. This straightforward procedure facilitates a quick and visual representation of the sample data's spatial distribution within Vadodara district, allowing for easy interpretation and analysis of the geographic patterns embedded in the provided dataset.

❖ Sample Raster Data

- Open QGIS and load the normalized raster images (NDVI, DEM, LST) into the project.
- Import the sample point file containing coordinates (X, Y) onto the QGIS canvas. You can use a CSV file with latitude and longitude columns to represent your sample points.
- Use the "Sample Raster Values" tool in QGIS to extract raster values for each sample point.
- Go to the Processing toolbox.
- Search for "Sample Raster Values" and select the tool.
- Choose the normalized NDVI, DEM, or LST raster layer as input.
- Set the point layer containing your sample points as the Input Point Layer.
- Specify an output layer for the sampled points.
- Repeat the above process for each normalized raster layer (NDVI, DEM, LST) separately, generating sampled points for each variable.
- Once sampling is completed for each raster layer, open the attribute table of the sample point layer in QGIS.
- Add new columns to the attribute table for NDVI, DEM, and LST.
- Populate these columns with the sampled values from the corresponding raster layers.
- Review the updated attribute table to ensure that NDVI, DEM, and LST values are successfully added to the sample point data.

❖ Stack the Images

- Import the normalized NDVI, DEM, and LST raster images into ERDAS Imagine.
- Display each raster image to ensure that they align spatially and have the same extent and resolution.
- In the ERDAS Imagine menu, go to raster then go to "Spectral.
- From the "Spectral" menu, select "Stack Layers" or a similar option that allows you to stack multiple layers.
- In the "Stack Layers" tool, select the normalized NDVI, DEM, and LST raster layers as input.
- Specify the output configuration, including the desired output file name and location.
- Arrange the bands in the desired order to ensure consistency. Typically, the order might be NDVI, DEM, and LST.

- Execute the Raster Stack tool to combine the individual raster layers into a single multiband image.
- Save the stacked image in a format of TIFF.

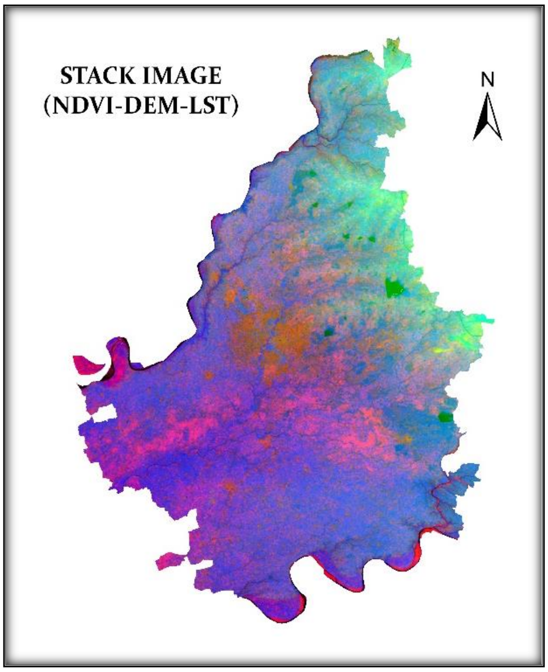

FIG : 3 Stack Image

❖ Generate soil nutrient mapping by using machine learning

By using python, we are building a model to predict soil nutrient levels (K20_kg_ha) using a **Linear Regression model**. We train the model using a dataset containing features like NDVI, DEM, and LST. The trained model is then applied to a new raster image, where NDVI, DEM, and LST bands are normalized using MinMaxScaler. The predictions are stored in a DataFrame, and the results are converted into a new raster image for regression analysis. The code also includes model evaluation metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to assess the model's performance on a test set. The final predicted EC raster image is visualized using Matplotlib, providing insights into the spatial distribution of predicted soil nutrient levels.

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - y_p)^2}{n}}$$

$$\text{MAE} = \frac{|(y_i - y_p)|}{n}$$

y_i = actual value
y_p = predicted value
n = number of observations/rows

FIG : 4 MAE & RMSE Equation

❖ Clipping the generated mapping TIFF file

- ➢ Use the "Add Raster Layer" button or menu option to load your TIFF file into QGIS.
- ➢ Similarly, load the Vadodara shapefile into QGIS using the "Add Vector Layer" button or menu option.
- ➢ Ensure that both the TIFF file and the Vadodara shapefile are visible in the Layers panel. Arrange the layers so that the shapefile is on top.
- ➢ Use the "Clip Raster by Mask Layer" tool to clip the TIFF file with the Vadodara shapefile.
- ➢ Go to Raster -> Extraction -> Clip Raster by Mask Layer.
- ➢ Select the input raster layer (Soil nutrient mapping TIFF file).
- ➢ Choose the mask layer (Vadodara shapefile).
- ➢ Specify the output file for the clipped raster.
- ➢ Click on the "Run" button to execute the clip operation.
- ➢ Save or export the clipped raster as a new GeoTIFF file.

❖ Generate the Map

Utilizing ArcGIS Pro, a map was generated from the TIFF file with essential elements for clarity. The map features a north arrow for orientation, a legend for data interpretation, latitude-longitude markings for precise location context, and a scale for distance comprehension. Finally, the map was exported into a user-friendly format, either JPG or PDF, ensuring accessibility and seamless sharing of the spatial information presented.

4. RESULT & CONCLUSION

4.1 Potassium Mapping

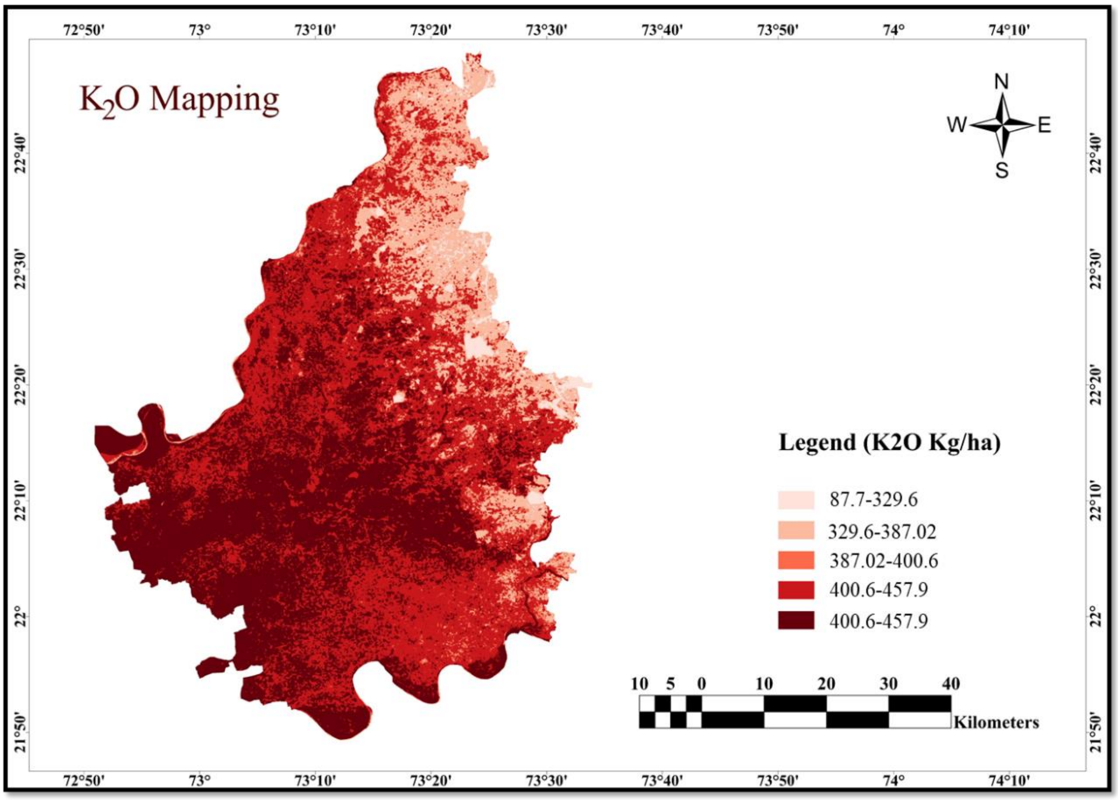

FIG : 5 Potassium Mapping

K Level	**Village**
87.7-329.6	Ranipura
	Tulsigam
	Udalpur
0.83-1.5	Nada
	Borbar

4.2 EC Mapping

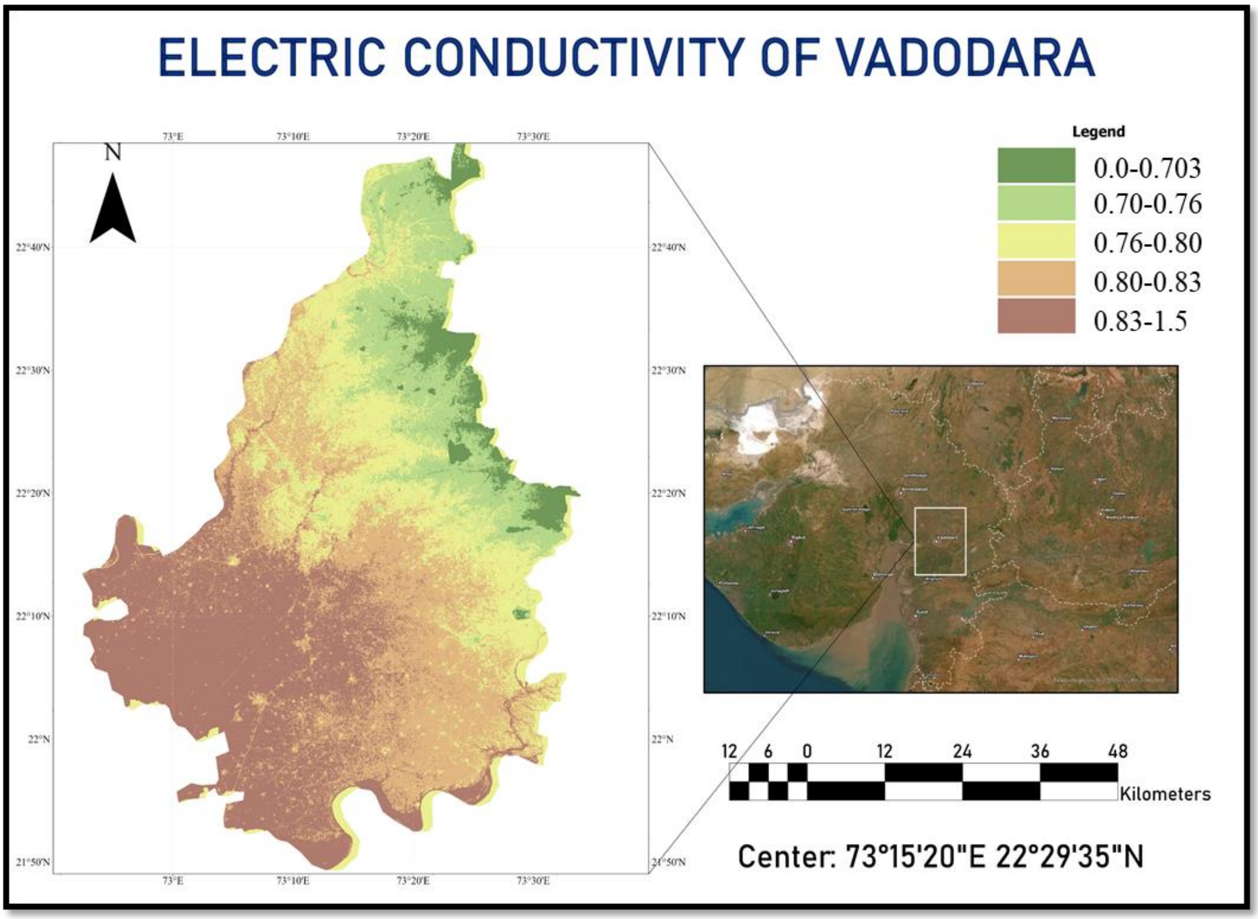

FIG : 6 EC Mapping

EC Level (dS/m)	**Village**
0.0-0.703	Tulsipura
	Gambhipur
	Vejpur
0.83-1.5	Pura
	Urad

Please note that this is a hypothetical scenario based on our provided information, and the accuracy of such mapping would depend on the availability and quality of data used for the machine learning algorithm.

APPENDEX

1. Download the data (NDVI,DEM,LST)

```
//DOWNLOAD THE NDVI DATA
Map.addLayer(dis_vad)
Map.centerObject(dis_vad)
var dataset = ee.ImageCollection("LANDSAT/LC08/C01/T1_32DAY_NDVI")
                .filter(ee.Filter.date('2017-01-01', '2017-12-31'));
var ndvi = dataset.select('NDVI');
var ndvi_01=ndvi.median().clip(dis_vad);
var ndviVis = {
  min: 0,
  max: 1,
  palette: [
    'ffffff', 'ce7e45', 'df923d', 'f1b555', 'fcd163', '99b718', '74a901',
    '66a000', '529400', '3e8601', '207401', '056201', '004c00', '023b01',
    '012e01', '011d01', '011301' ],
};
print(ndvi_01.projection());
Map.setCenter(6.746, 46.529, 6);
Map.addLayer(ndvi_01.clip(dis_vad), ndviVis, 'NDVI');
Export.image.toDrive({
  image: ndvi_01,
  description: 'imageToDriveExample_transform',
  crs:'EPSG:4326',
  region: dis_vad
});

//DOWNLOAD THE DEM DATA
var elevationDataset = ee.Image("CGIAR/SRTM90_V4");
var clippedElevation = elevationDataset.clip(dis_vad);
var elevationVis = {
```

```
  min: 0,  // Adjust the minimum value as needed
  max: 60,  // Adjust the maximum value as needed
  palette: ['0000ff', '00ff00', 'ff0000']  // Adjust the colors as needed
};
Map.setCenter(6.746, 46.529, 6);
Map.addLayer(clippedElevation, elevationVis, 'Elevation');
Export.image.toDrive({
  image: clippedElevation,
  description: 'elevation_image',
  crs: 'EPSG:4326',  // Set the desired CRS (e.g., WGS84)
  region: dis_vad,
  maxPixels: 1e10  // Adjust as needed
});

// DOWNLOAD THE LST DATA
var landsatCollection = ee.ImageCollection("LANDSAT/LC08/C01/T1_TOA")
    .filterBounds(dis_vad)
    .filterDate('2017-01-01', '2017-12-31');
var lstImage = landsatCollection.select('B10');
var clippedLST = lstImage.clip(dis_vad);
var lstVis = {
  min: 273,  // Adjust the minimum value as needed
  max: 313,  // Adjust the maximum value as needed
  palette: ['blue', 'cyan', 'green', 'yellow', 'red']  // Adjust the colors as needed
};
Map.setCenter(6.746, 46.529, 6);
Map.addLayer(clippedLST, lstVis, 'Land Surface Temperature');
Export.image.toDrive({
  image: clippedLST,
  description: 'lst_image',
  crs: 'EPSG:4326',
  region: dis_vad,
  maxPixels: 1e10
});
```

2. Generate soil nutrient map by using linear regression

```
import rasterio
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.preprocessing import MinMaxScaler
import joblib
from google.colab import drive
drive.mount('/content/gdrive')

# Load your dataset for training
file_path = "/content/SAMPLE_RASTER_MERGE_TSET.csv"
df = pd.read_csv(file_path)
Y_PRE = df[['K20_kg_ha_']]

# Extract the relevant columns for training and prediction
features = ['NDVI', 'DEM', 'LST']
target = Y_PRE

# Split the data into features (X) and target (y)
X = df[features]
y = target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
# Save the trained model to a file
model_save_path = "/content/trained_ec_model.joblib"
joblib.dump(model, model_save_path)

# Load the raster image file
file_path = "/content/gdrive/MyDrive/NR_STACK_MERGE_TEST.tif"
src = rasterio.open(file_path)

# Read the image data for bands 1, 2, and 3
band1_data = src.read(1)
band2_data = src.read(2)
band3_data = src.read(3)

# Reshape the data to 1D arrays
band1_flat = band1_data.flatten()
band2_flat = band2_data.flatten()
band3_flat = band3_data.flatten()

# Normalize the bands using MinMaxScaler
scaler = MinMaxScaler()

band1_normalized = scaler.fit_transform(band1_flat.reshape(-1, 1)).flatten()
band2_normalized = scaler.fit_transform(band2_flat.reshape(-1, 1)).flatten()
band3_normalized = scaler.fit_transform(band3_flat.reshape(-1, 1)).flatten()

# Create a DataFrame using pandas
new_data = pd.DataFrame({
    'NDVI': band1_normalized,
    'DEM': band2_normalized,
    'LST': band3_normalized,
})

# Load the trained model
```

```
loaded_model = joblib.load(model_save_path)

# Make predictions on the new data
new_predictions = loaded_model.predict(new_data)

# Create a DataFrame using pandas
df_classification = pd.DataFrame({
    'NDVI': list(band1_normalized),
    'DEM': list(band2_normalized),
    'LST': list(band3_normalized),
    'EC': list(new_predictions)
})

# Make predictions on the raster image for regression
X_raster_regression = df_classification[['NDVI', 'DEM', 'LST']]
df_classification['predicted_EC'] = loaded_model.predict(X_raster_regression)

# Reshape the predicted values to create a new raster image for regression
predicted_image_regression =
df_classification['predicted_EC'].values.reshape(band1_data.shape)
modified_list = [[max(0, value) for value in row] for row in predicted_image_regression]

# Specify the path to save the predicted EC raster image
predicted_ec_image_path = "/content/predicted_K20_kg_ha_without_NR_image.tif"
# print(predicted_image_regression)
# input_array = np.array(predicted_image_regression)

# Evaluate the model on the test set
y_pred = model.predict(X_test)
rmse = mean_squared_error(y_test, y_pred, squared=False)
mae = mean_absolute_error(y_test, y_pred)
print(f"Test RMSE: {rmse:.4f}")
print(f"Test MAE: {mae:.4f}")
```

```
# Flatten the list of lists
flat_list = [item for sublist in modified_list for item in sublist]
# Find the minimum and maximum values
min_value = min(flat_list)
max_value = max(flat_list)

print("Minimum value:", min_value)
print("Maximum value:", max_value)

# Get the metadata from the original raster file
meta = src.meta

# Update metadata for the new raster file
meta.update(dtype='float32', count=1)

# Convert the list to a NumPy array
modified_array = np.array(modified_list, dtype=np.float32)

# Open the raster file for writing
with rasterio.open(predicted_ec_image_path, 'w', **meta) as dst:
    # Write the NumPy array to the raster file
    dst.write(modified_array, 1)

# Display the raster image for regression using Matplotlib
plt.figure(figsize=(8, 8))
plt.imshow(modified_list, cmap='viridis')  # You can change the colormap (cmap) as
needed
plt.colorbar()
plt.title("Predicted K20 Image (Regression)")
plt.axis('off')
plt.show()

# Close the raster files
src.close()
```

BIBLIOGRAPHY

Khosravani, P., Baghernejad, M., Moosavi, A. A., & FallahShamsi, S. R. (2023). Digital mapping to extrapolate the selected soil fertility attributes in calcareous soils of a semiarid region in Iran. Journal of Soils and Sediments, 1-23.

Malone, B. P., Minasny, B., Odgers, N. P., & McBratney, A. B. (2014). Using model averaging to combine soil property rasters from legacy soil maps and from point data. Geoderma, 232, 34-44.

Mahmoudabadi, E., Karimi, A., Haghnia, G. H., & Sepehr, A. (2017). Digital soil mapping using remote sensing indices, terrain attributes, and vegetation features in the rangelands of northeastern Iran. Environmental monitoring and assessment, 189, 1-20.

Wu, Z., Chen, Y., Zhu, Y., Feng, X., Ou, J., Li, G., ... & Yan, Q. (2023). Mapping Soil Organic Carbon in Floodplain Farmland: Implications of Effective Range of Environmental Variables. Land, 12(6), 1198.

Wilson, S. G., Rees, G. L., & O'Geen, T. A Super Learner Ensemble to Map Potassium Fixation in California Vineyard Soils. Available at SSRN 4618992.

https://buildmedia.readthedocs.org/media/pdf/rasterio/latest/rasterio.pdf