# Landsat Soil Classification Using Machine Learning Models – Visualizations and Architecture Selection

*AI for Engineers (S-22) – Course Project*

*Visvesvaraya National Institute of Technology, Nagpur*


1. **Abhishek Mhamane (BT18CIV057)**
2. **Darshan Dhote (BT18CME070)**
3. **Nerella Akhil (BT18MEC076)**
4. **Chilagani Chaitanya Chandra (BT18MEC132)**
5. **Jillella Sai Charan Reddy (BT18ECE123)**

***Under the Guidance of***
Dr. Tushar Gohil

**Department of Mechanical Engineering**

**Visvesvaraya National Institute of Technology**

**Nagpur 440010 (India)**


**April, 2022**

# Table of Contents

# Introduction

Remote Sensing is the acquisition of information about an object or phenomenon without making physical contact with the object, in contrast to in situ or on-site observation. Remote Sensing is used in numerous fields, including geography, agriculture, land surveying, meteorology, hydrology, oceanography, military intelligence, urban planning and humanitarian application. The term “remote sensing” generally refers to the use of satellite or aircraft-based sensor technologies to detect and classify objects on Earth. There are two types of remote – active and passive remote sensing. There are different data products available from different space agencies -

*Table 1: Satellite Data Products*

| Space Agency / Organizations | Data Product / Sensors / Mission |
|---|---|
| ISRO | LISS-3, CartoSAT, AWiFS, RiSAT, etc, |
| NASA | Landsat, MODIS, ICEsat, SRTM DEM, GRACE, etc. |
| ESA | Sentinal, Copernicus, SPOT, GOCE, etc. |
| Private Companies | IKONOS |

The Landsat program is the longest-running enterprise for acquisition of satellite imagery of Earth. It is a joint NASA/USGS program. The latest addition being the launch of Landsat 9 on 21 September 2021. The images, archived in the United States and at Landsat receiving stations around the world, are a unique resource for global change research and applications in agriculture, cartography, geology, forestry, regional planning, surveillance and education, and can be viewed through the U.S. Geological Survey (USGS) "EarthExplorer" website.

Application of AI in Remote Sensing is a growing field, (1) land cover analysis (including forest clear-cut monitoring and vegetation change detection); (2) map-assisted photointerpretation; (3) image segmentation; (4) hyperspectral image analysis; (5) cartographic feature extraction; (6) analysis of geological structures; and (7) structural analysis of complex aerial photographs. All of the above problems can be solved using AI based algorithms.

# Literature Review

The UCI 'satellite' dataset has 4435 examples for training and 2000 for testing. The problem is to recognize 6 different classes of objects using satellite image data. The efficient partition algorithm calculated 8.2% errors for the test set. The Associative Neural Network (ASNN) algorithm further improved this result to 7.8% test set errors, that is the best published result for this set[1].

| Algorithm | Maximum Storage | Time (sec.) Train | Time (sec.) Test | Error Rate Train | Error Rate Test | Rank |
|---|---|---|---|---|---|---|
| Discrim | 254 | 67.8 | 11.9 | 0.149 | 0.171 | 19 |
| Quadisc | 364 | 157.0 | 52.9 | 0.106 | 0.155 | 14 |
| Logdisc | 1205 | 4414.1 | 41.2 | 0.119 | 0.163 | 17 |
| SMART | 244 | 27376.2 | 10.8 | 0.123 | 0.159 | 16 |
| ALLOC80 | 244 | 63840.2 | 28756.5 | 0.036 | 0.132 | 5 |
| k-NN | 180 | 2104.9 | 944.1 | 0.089 | 0.094 | 1 |
| CASTLE | * | 75.0 | 80.0 | 0.186 | 0.194 | 21 |
| CART | 253 | 329.9 | 14.2 | 0.079 | 0.138 | 6 |
| IndCART | 819 | 2109.2 | 9.2 | 0.023 | 0.138 | 6 |
| NewID | 1800 | 226.0 | 53.0 | 0.067 | 0.150 | 10 |
| $AC^2$ | * | 8244.0 | 17403.0 | * | 0.157 | 15 |
| Baytree | 161 | 247.8 | 10.2 | 0.020 | 0.147 | 9 |
| NaiveBay | 133 | 75.1 | 16.5 | 0.308 | 0.287 | 22 |
| CN2 | 682 | 1664.0 | 35.8 | 0.010 | 0.150 | 10 |
| $C4.5^4$ | 1150 | 434.0 | 1.0 | 0.040 | 0.150 | 10 |
| ITrule | FD | FD | FD | FD | FD | |
| Cal5 | 412 | 764.0 | 7.2 | 0.125 | 0.151 | 13 |
| Kohonen | * | 12627.0 | 129.0 | 0.101 | 0.179 | 20 |
| DIPOL92 | 293 | 764.3 | 110.7 | 0.051 | 0.111 | 3 |
| Backprop | 469 | 72494.5 | 52.6 | 0.112 | 0.139 | 8 |
| RBF | 195 | 564.2 | 74.1 | 0.111 | 0.121 | 4 |
| LVQ | 227 | 1273.2 | 44.2 | 0.048 | 0.105 | 2 |
| Cascade | 1210 | 7180.0 | 1.0 | 0.112 | 0.163 | 17 |
| Default | * | * | * | 0.758 | 0.769 | 23 |

*Figure 1: Performance of different Algorithms on Landsat Dataset[2]*

In the satellite image dataset k-NN performs best[2]. Their success suggests that all the attributes are equally scaled and equally important. There appears to be little to choose between any of the other algorithms, except that Naive Bayes does badly (and its close relative CASTLE also does relatively badly). The Decision Tree algorithms perform at about the same level, with CART giving the best result using 66 nodes. This dataset has the highest correlation between attributes(corr.abs = 0.5977). This may partly explain the failure of Naïve Bayes(assumes attributes are conditionally independent), and CASTLE (confused if several attributes contain equal amounts of information)[2].

The examples were created using a 3 x 3 neighbourhood so it is no surprise that there is a very large correlation amongst the 36 variables. The results from CASTLE suggest that only three of the variables for the centre pixel are necessary to classify the observation. However, other algorithms found a significant improvement when information from the neighbouring pixels was used[2].

Many researchers also focused on data processing so as to generalize the model and remove the effect on outliers on achieving convergence. The representation and quality of the instance data is first and foremost. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult [3]. Data pre-processing includes data cleaning, normalization, transformation, feature extraction and selection, etc. The product of data pre-processing is the final training set.

In real-world data, the representation of data often uses too many features, but only a few of them may be related to the target concept. There may be redundancy, certain features are correlated so it is not necessary to include all of them in modelling. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. Normalization is a "scaling down" transformation of the features. Within a feature there is often a large difference between the maximum and minimum values. hen normalization is performed the value magnitudes and scaled to appreciably low values. This is important for many neural network and k-Nearest Neighbourhood algorithms. Two common normalizations (1) min-max normalization and (2) z-score normalization[3]. Further the imbalance in data can also affect the generalization achieved by the model. Data outliers can be identified if the data fall beyond the range b/w Q1 – 1.5(Q3-Q1) and Q3 + 1.5(Q3-Q1), where Q1 and Q3 represent the first and third quartiles respectively.[4].

# Objective

1. Develop understanding about different Machine Learning techniques for Classification.
2. Develop a Neural Network based classification model.
3. Visualize the inner workings of Neural Network "Black-Box" nature, to develop better understanding.
4. Find optimal architecture for this problem.
5. Identify potential shortcomings in the approach utilized and suggest future improvements.

# Dataset - Statlog (Landsat Satellite) Data Set

## Introduction

The original Landsat data for this database was generated from data purchased from NASA by the Australian Centre for Remote Sensing, and used for research at the University of New South Wales. The sample database was generated taking a small section (82 rows and 100 columns) from the original data. The classification for each pixel was performed on the basis of an actual site visit by Ms. Karen Hall, when working for Professor John A. Richards, at the Centre for Remote Sensing. The database is a (tiny) sub-area of a scene, consisting of 82x100 pixels, each pixel covering an area on the ground of approximately 80*80 metres. The information given for each pixel consists of the class value and the intensities in four spectral bands(from the green, red, and infra-red regions of the spectrum).

Each line of data corresponds to a 3 x 3 square neighbourhood of pixels completely contained within the 82 x 100 sub-area. Thus, each line contains the four spectral bands of each of the 9 pixels in the 3 x 3 neighbourhood and the class of the central pixel which was one of: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, very damp grey soil. The dataset was divided into training data of 4435 examples and test data of 2000 examples.

The dataset is hosted at –
https://archive.ics.uci.edu/ml/datasets/Statlog+%28Landsat+Satellite%29

```
train_df.head()
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 84 | 102 | 102 | 83 | 80 | 102 | 102 | 79 | 84 | 94 | ... | 87 | 84 | 99 | 104 | 79 | 84 | 99 | 104 | 79 | 3 |
| 1 | 80 | 102 | 102 | 79 | 84 | 94 | 102 | 79 | 80 | 94 | ... | 79 | 84 | 99 | 104 | 79 | 84 | 103 | 104 | 79 | 3 |
| 2 | 84 | 94 | 102 | 79 | 80 | 94 | 98 | 76 | 80 | 102 | ... | 79 | 84 | 103 | 104 | 79 | 79 | 107 | 109 | 87 | 3 |
| 3 | 80 | 94 | 98 | 76 | 80 | 102 | 102 | 79 | 76 | 102 | ... | 79 | 79 | 107 | 109 | 87 | 79 | 107 | 109 | 87 | 3 |
| 4 | 76 | 102 | 106 | 83 | 76 | 102 | 106 | 87 | 80 | 98 | ... | 87 | 79 | 103 | 104 | 83 | 79 | 103 | 104 | 79 | 3 |

5 rows × 37 columns

```
test_df.head()
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 80 | 102 | 102 | 79 | 76 | 102 | 102 | 79 | 76 | 102 | ... | 87 | 79 | 107 | 109 | 87 | 79 | 107 | 113 | 87 | 3 |
| 1 | 76 | 102 | 102 | 79 | 76 | 102 | 106 | 83 | 76 | 102 | ... | 87 | 79 | 107 | 113 | 87 | 79 | 103 | 104 | 83 | 3 |
| 2 | 80 | 98 | 106 | 79 | 76 | 94 | 102 | 76 | 76 | 94 | ... | 79 | 79 | 95 | 100 | 79 | 79 | 95 | 96 | 75 | 4 |
| 3 | 76 | 94 | 102 | 76 | 76 | 94 | 102 | 76 | 76 | 94 | ... | 79 | 79 | 95 | 96 | 75 | 79 | 95 | 100 | 75 | 4 |
| 4 | 76 | 94 | 102 | 76 | 76 | 94 | 102 | 76 | 76 | 89 | ... | 75 | 79 | 95 | 100 | 75 | 75 | 95 | 100 | 79 | 4 |

*Figure 2: Raw Data*

## Exploring Dataset

The dataset has 6 soil classes. The frequency of each soil class is not same. Soil type 1 and 7 are over represented with 1072 and 1038 examples respectively. Soil type 3, 2, 5 and 4 have 959, 479, 470 and 415 examples respectively. Thus, the model is going to be biased towards soil type 1 and 7.

The 3x3 neighbourhood data representation, is a lot correlated data. For ease of data-handling only central pixels are used. Thus, reducing input feature size from 36 to just 4. The spectral band data is summarized in the table below.

| | b1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | count | mean | std | min | 25% | 50% | 75% | max |
| soil_type | | | | | | | | |
| 1 | 1072 | 62.82556 | 8.021469 | 46 | 57 | 63 | 68 | 97 |
| 2 | 479 | 48.839248 | 7.570674 | 40 | 44 | 46 | 50 | 78 |
| 3 | 959 | 87.477581 | 5.041505 | 70 | 84 | 88 | 92 | 104 |
| 4 | 415 | 77.409639 | 5.543931 | 64 | 74 | 78 | 82 | 92 |
| 5 | 470 | 59.589362 | 6.087449 | 44 | 56 | 59 | 63 | 82 |
| 7 | 1038 | 69.012524 | 5.382105 | 52 | 66 | 68 | 71 | 88 |

| | b2 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| soil_type | count | mean | std | min | 25% | 50% | 75% | max |
| 1 | 1072 | 95.293843 | 14.548237 | 61 | 83 | 99 | 108 | 121 |
| 2 | 479 | 39.914405 | 13.483252 | 27 | 32 | 34 | 42 | 88 |
| 3 | 959 | 105.494265 | 6.869171 | 83 | 100 | 106 | 111 | 130 |
| 4 | 415 | 90.944578 | 8.15871 | 66 | 87 | 91 | 95 | 112 |
| 5 | 470 | 62.265957 | 11.637356 | 43 | 54 | 60 | 69 | 99 |
| 7 | 1038 | 77.421965 | 7.687055 | 60 | 72 | 77 | 81 | 103 |

| | b3 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| soil_type | count | mean | std | min | 25% | 50% | 75% | max |
| 1 | 1072 | 108.123134 | 12.636916 | 74 | 98 | 111 | 118.25 | 135 |
| 2 | 479 | 113.889353 | 12.641098 | 82 | 104 | 114 | 124 | 139 |
| 3 | 959 | 110.595412 | 7.231846 | 85 | 105 | 110 | 114 | 139 |
| 4 | 415 | 95.614458 | 7.910782 | 68 | 90 | 96 | 100 | 119 |
| 5 | 470 | 83.023404 | 12.570377 | 56 | 74 | 82 | 89 | 122 |
| 7 | 1038 | 81.592486 | 8.741692 | 62 | 75 | 80 | 86 | 114 |

| | b4 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| soil_type | count | mean | std | min | 25% | 50% | 75% | max |
| 1 | 1072 | 88.600746 | 8.824099 | 65 | 81 | 90 | 96 | 104 |
| 2 | 479 | 118.311065 | 19.293952 | 67 | 105 | 122 | 133 | 157 |
| 3 | 959 | 87.466111 | 6.049263 | 59 | 83 | 87 | 92 | 109 |
| 4 | 415 | 75.354217 | 6.532884 | 59 | 70 | 75 | 79 | 94 |
| 5 | 470 | 69.953191 | 13.125357 | 34 | 61 | 70 | 78 | 100 |
| 7 | 1038 | 64.125241 | 7.361824 | 48 | 58 | 63 | 68 | 90 |

*Figure 3: Soil Statistics band and type wise*

Using this data spectral reflectance curves were plotted. Spectral Reflectance curves are unique for each feature, and are of vital importance in classification tasks.

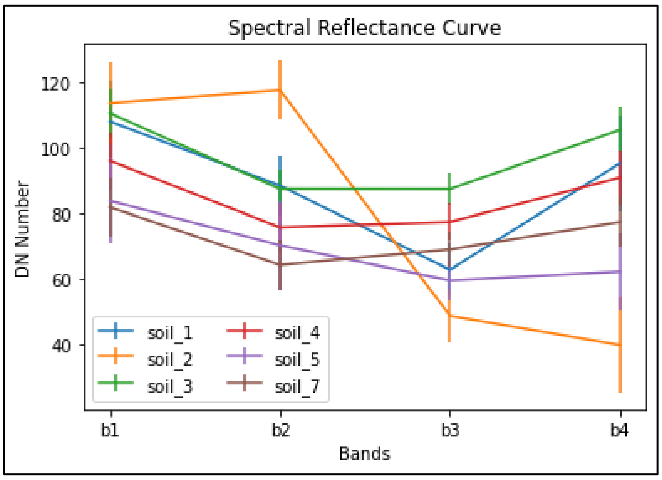


*Figure 4: Spectral Reflectance Curve*

Further the data could be visualized through the use of Box-plots and scatter plots. Box plots help in detecting the outliers, by presenting the $25^{th}$ and $75^{th}$ percentile along with the inter quartile range. Scatter plots help understand the presence of clustering, which often is the case in such classification problems. The plots are shown below:

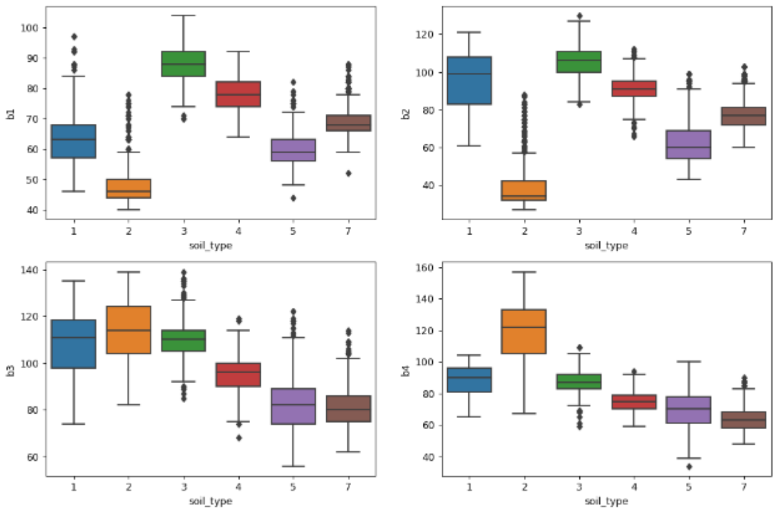


*Figure 5: Box Plot of training data*

From Box-plot and pair plot it can be observed that there is natural clustering present in the data but there are also many outliers. Outlier removal is required. Outlier removal is done are described earlier[4].

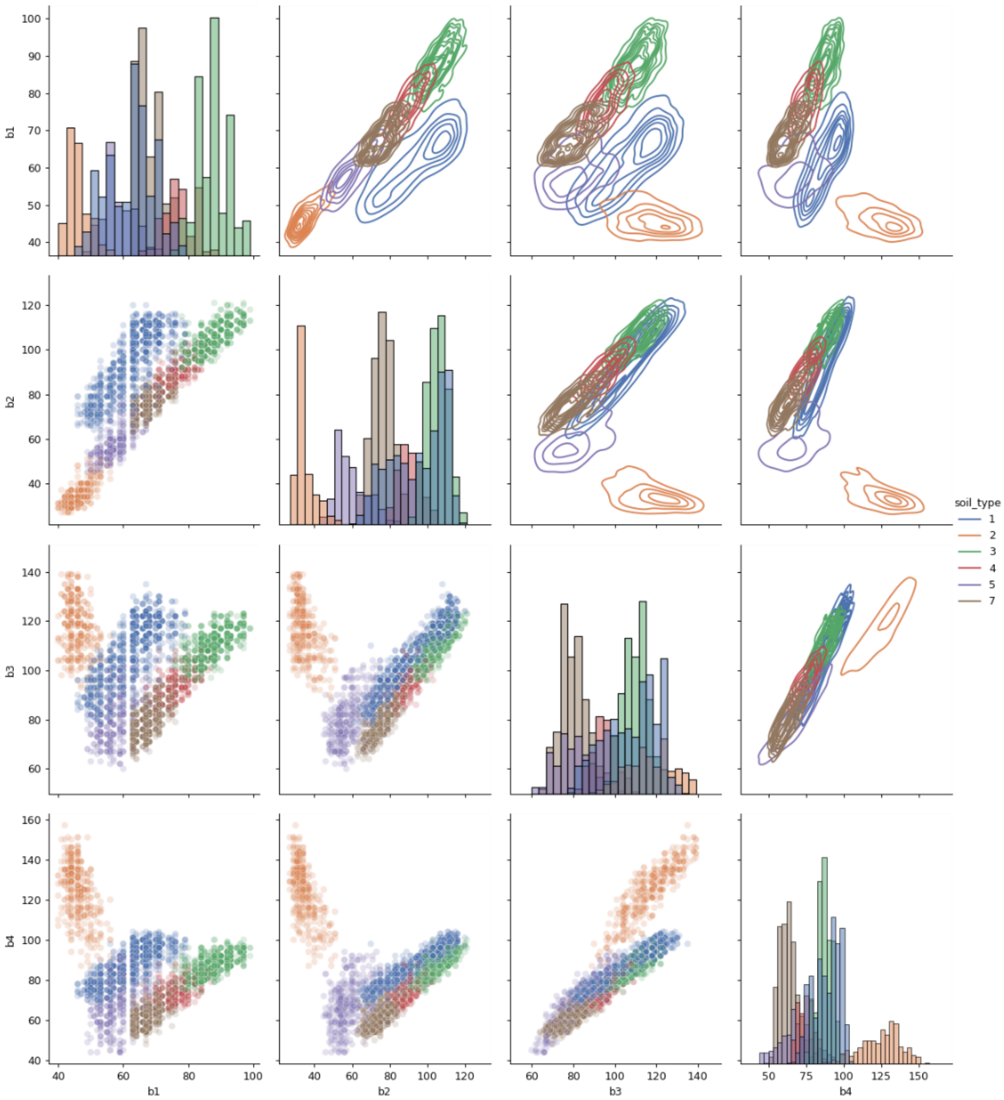


*Figure 6: Pair Plot of training data*

From the scatter plot it can also be observed that some band combinations are really helpful such as – “b1 and b3” and “b1 and b2”. The cluster identification is much easier for these band combinations.

# Using Machine Learning Algorithms

## KNN Classifier

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. Accuracy achieved ~70%.

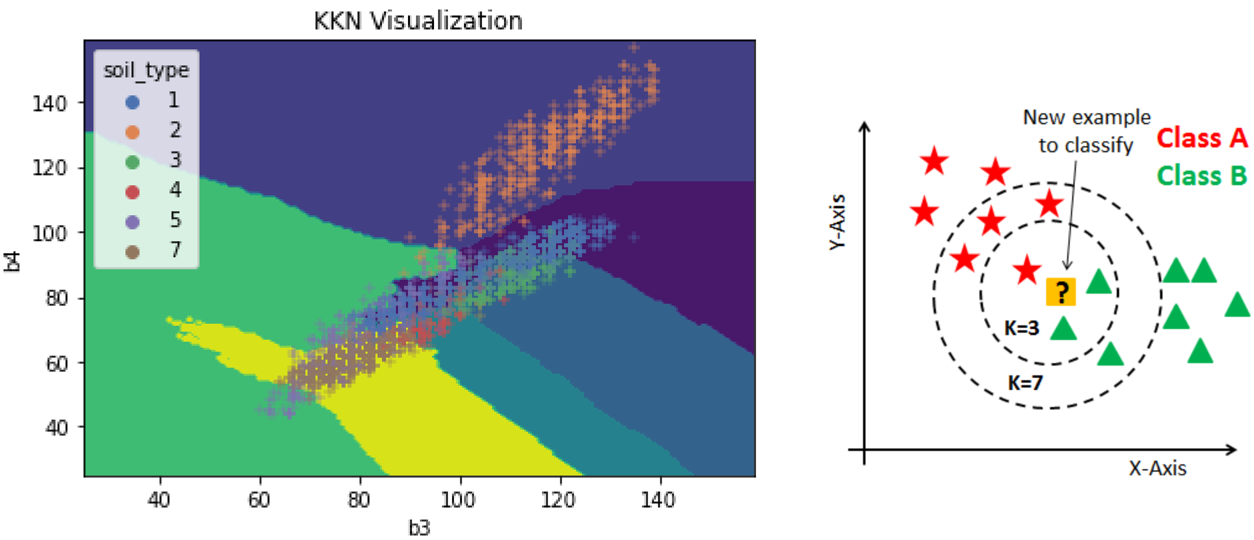


*Figure 7: Visualization of decision boundaries - KNN Classifier*

## Nearest Centroid Classifier

The Nearest Centroid classifier is arguably the simplest Classification algorithm in Machine Learning. The Nearest Centroid classifier works on a simple principle : Given a data point (observation), the Nearest Centroid classifier simply assign it the label (class) of the training sample whose mean or centroid is closest to it. The Nearest Centroid classifier is somewhat similar to the K-Nearest Neighbours classifier. Nearest Centroid accuracy achieved ~82%.

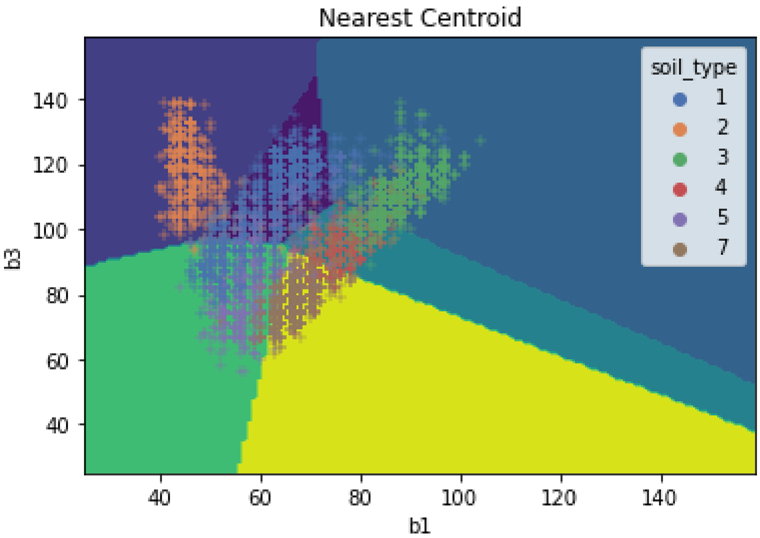


*Figure 8: Nearest Centroid Clasifier Decision Boundries*

## Neural Network Classifier

A neural network consists of units (neurons), arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer. Generally the networks are defined to be feed-forward: a unit feeds its output to all the units on the next layer, but there is no feedback to the previous layer. Weightings are applied to the signals passing from one unit to another, and it is these weightings which are tuned in the training phase to adapt a neural network to the particular problem at hand. This is the learning phase.

Different Activation functions have different influence on decision boundaries even for same architecture. Classification accuracy achieved ~82%. These are shown below:

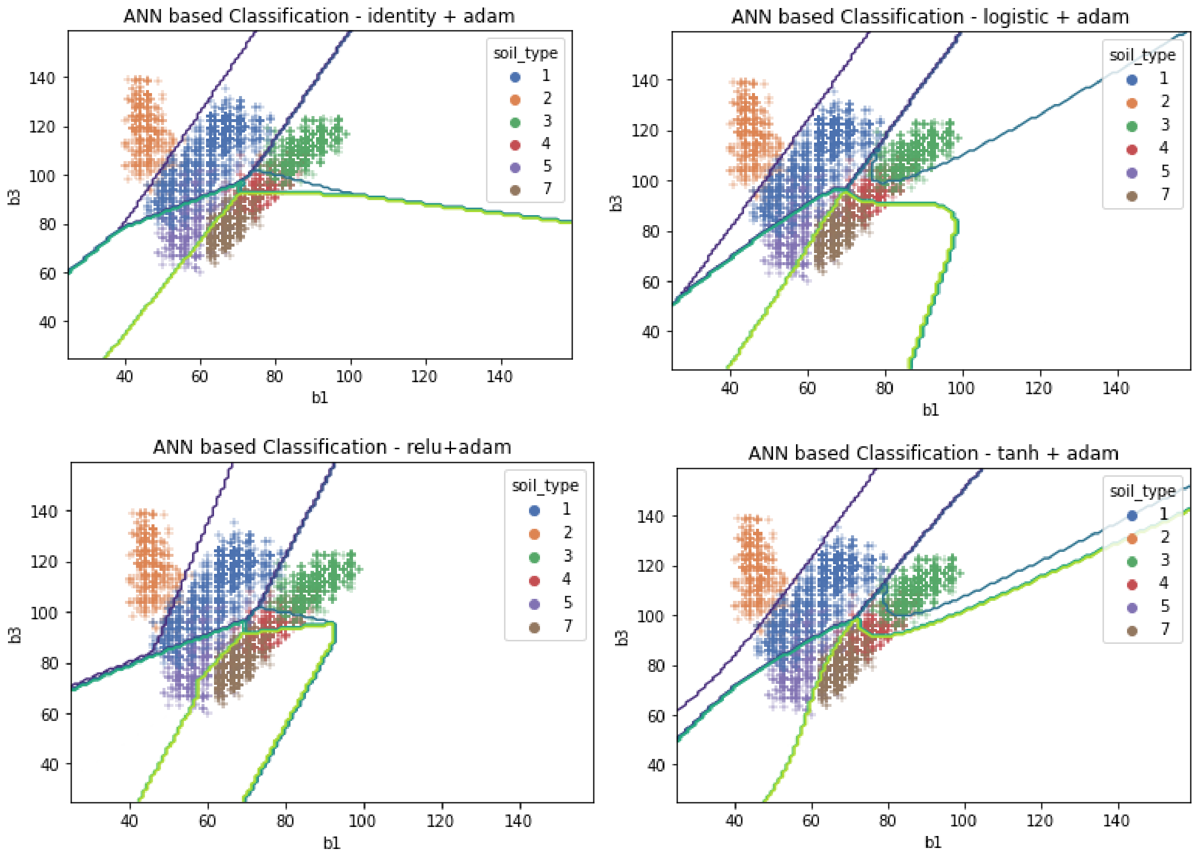


*Figure 9: ANN based classifier*

The identity activation function generates linear decision boundaries. Whereas, ReLU being piecewise linear but provides non-linear boundaries. Tanh generates highly non-linear decision boundaries. Soil type 4 is somewhat difficult to classify. This may be due to considering only 2 features.

# Architecture – Selection

In order to find the optimal neural network architecture, different number of hidden layers with different number of neurons is tried. Accuracy for each of the layer is found. This data is tabulatd below:

| Architecture (Hidden Layers) | Activation Function | Epochs | Optimizer | | Overall Clssification Accuracy | |
|---|---|---|---|---|---|---|
| | | | Optimizer | Learning Rate | Train Accuracy | Test Accuracy |
| 1 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.01 | 82.35% | 80.97% |
| 1 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.1 | 83.56% | 80.59% |
| 1 layer - 4 neurons | Tanh | 500 | AdagGrad | 0.01 | 65.40% | 65.60% |
| 1 layer - 4 neurons | Tanh | 500 | AdaGrad | 0.1 | 66.92% | 66.87% |
| | | | | | | |
| 2 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.01 | 69.20% | 66.37% |
| 2 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.1 | 80.96% | 77.32% |
| 2 layer - 4 neurons | Tanh | 500 | AdagGrad | 0.01 | 89.10% | 85.90% |
| 2 layer - 4 neurons | Tanh | 500 | AdaGrad | 0.1 | 82.50% | 80.42% |
| | | | | | | |
| 3 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.01 | 81.32% | 78.54% |
| 3 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.1 | 87.57% | 86.39% |
| 3 layer - 4 neurons | Tanh | 500 | AdagGrad | 0.01 | 65.55% | 64.16% |
| 3 layer - 4 neurons | Tanh | 500 | AdaGrad | 0.1 | 66.41% | 66.64% |
| | | | | | | |
| 4 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.01 | 50.48% | 47.34% |
| 4 layer - 4 neurons | ReLU | 500 | AdaGrad | 0.1 | 79.63% | 76.77% |
| 4 layer - 4 neurons | Tanh | 500 | AdagGrad | 0.01 | 57.23% | 55.92% |
| 4 layer - 4 neurons | Tanh | 500 | AdaGrad | 0.1 | 82.60% | 79.09% |
| | | | | | | |
| 5 layers - 4 neurons | ReLU | 500 | AdaGrad | 0.01 | 61.37% | 57.85% |
| 5 layers - 4 neurons | ReLU | 500 | AdaGrad | 0.1 | 78.64% | 75.38% |
| 5 layers - 4 neurons | Tanh | 500 | AdagGrad | 0.01 | 53.55% | 50.60% |
| 5 layers - 4 neurons | Tanh | 500 | AdaGrad | 0.1 | 81.82% | 78.54% |
| | | | | | | |
| | | | | | | |
| 1 layer - 6 neurons | ReLU | 500 | AdaGrad | 0.01 | 88.63% | 86.22% |
| 1 layer - 6 neurons | ReLU | 500 | AdaGrad | 0.1 | 88.07% | 84.45% |
| 1 layer - 6 neurons | Tanh | 500 | AdagGrad | 0.01 | 65.43% | 65.32% |
| 1 layer - 6 neurons | Tanh | 500 | AdaGrad | 0.1 | 67.02% | 66.81 |
| | | | | | | |
| 2 layer - 6 neurons | ReLU | 500 | AdaGrad | 0.01 | 83.21% | 81.91% |
| 2 layer - 6 neurons | ReLU | 500 | AdaGrad | 0.1 | 86.58% | 84.95% |
| 2 layer - 6 neurons | Tanh | 500 | AdagGrad | 0.01 | 91.65% | 88.66% |
| 2 layer - 6 neurons | Tanh | 500 | AdaGrad | 0.1 | 92.61% | 90.43% |
| | | | | | | |
| 3 layer - 6 neurons | ReLU | 500 | AdaGrad | 0.01 | 80.15% | 76.66% |
| 3 layer - 6 neurons | ReLU | 500 | AdaGrad | 0.1 | 56.28% | 53.87% |
| 3 layer - 6 neurons | Tanh | 500 | AdagGrad | 0.01 | 67.75% | 66.54% |
| 3 layer - 6 neurons | Tanh | 500 | AdaGrad | 0.1 | 92.23% | 90.26% |
| | | | | | | |
| 1 layer - 8 neurons | ReLU | 500 | AdaGrad | 0.01 | 85.65% | 82.13% |
| 1 layer - 8 neurons | ReLU | 500 | AdaGrad | 0.1 | 92.13% | 89.93% |
| 1 layer - 8 neurons | Tanh | 500 | AdagGrad | 0.01 | 89.48% | 87.16% |
| 1 layer - 8 neurons | Tanh | 500 | AdaGrad | 0.1 | 92.43% | 90.48% |
| | | | | | | |
| 2 layer - 8 neurons | ReLU | 500 | AdaGrad | 0.01 | 85.02% | 82.52% |
| 2 layer - 8 neurons | ReLU | 500 | AdaGrad | 0.1 | 84.72% | 81.13% |
| 2 layer - 8 neurons | Tanh | 500 | AdagGrad | 0.01 | 71.81% | 70.96% |
| 2 layer - 8 neurons | Tanh | 500 | AdaGrad | 0.1 | 75.77% | 76.27% |
| | | | | | | |
| 3 layer - 8 neurons | ReLU | 500 | AdaGrad | 0.01 | 75.52% | 74.06% |
| 3 layer - 8 neurons | ReLU | 500 | AdaGrad | 0.1 | 90.42% | 86.78% |
| 3 layer - 8 neurons | Tanh | 500 | AdagGrad | 0.01 | 81.77% | 79.70% |
| 3 layer - 8 neurons | Tanh | 500 | AdaGrad | 0.1 | 82.60% | 80.53% |

*Figure 10: Architecture Selection*

# Results and Discussion

The traditional multi-layer neural network (MLP) is a memoryless approach. This means that after training is complete all information about the input patterns is stored in the neural network weights and input data are no longer needed, i.e. there is no explicit storage of any presented example in the system. Contrary to that, such methods as the k-nearest-neighbour's (KNN) represent the memory-based approaches. These approaches keep in memory the entire database of examples and their predictions are based on some local approximation of the stored examples. The neural networks can be considered global models, while the other two approaches are usually considered local models.

# References

[1] I. V. TETKO, “Associative Neural Network,” *Neural Process. Lett.*, vol. 16, pp. 187–199, 2002, doi: 10.1023/A:1019903710291.

[2] J. F. E. IV, D. Michie, D. J. Spiegelhalter, and C. C. Taylor, “Machine Learning, Neural, and Statistical Classification.,” *J. Am. Stat. Assoc.*, vol. 91, no. 433, p. 436, 1996, doi: 10.2307/2291432.

[3] S. B. Kotsiantis and D. Kanellopoulos, “Data preprocessing for supervised leaning,” *Int. J. …*, vol. 1, no. 2, pp. 1–7, 2006, doi: 10.1080/02331931003692557.

[4] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, “A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data,” *Front. Energy Res.*, vol. 9, no. March, pp. 1–17, 2021, doi: 10.3389/fenrg.2021.652801.