SOIL ANALYSIS AND CROP FERTILITY PREDICTION USING MACHINE LEARNING

Jagdeep Yadav,
Student, Department of Information Technology,
Vivekanand Education Society Institute of Technology,
Chembur, Mumbai, India
2018.jagdeep.yadav@ves.ac.in
Shalu Chopra,
Assistant Professor, Department of Information Technology,
Vivekanand Education Society Institute of Technology,
Chembur, Mumbai,India
shaluchopra@ves.ac.in
Vijayalakshmi M
Professor, Department of Information Technology,
Vivekanand Education Society Institute of Technology,
Chembur, Mumbai,India
m.vijayalakshmi@ves.ac.in

Publication History
Research Article | Open Access
Peer-review: Double-blind Peer-reviewed
Article ID: IJIRAE/RS/Vol.08/Issue03/MRAE10082
Received: 02, March 2021
Accepted: 20, March 2021
Published Online: 07, April 2021
Volume 2021 | Article ID MRAE10082 | https://doi.org/10.26562/ijirae.2021.v0803.003
Jagdeep,Shalu,Viyayalakshmi(2021). Soil Analysis and Crop Fertility Prediction using Machine Learning. IJIRAE::International Journal of Innovative Research in Advanced Engineering,Vol: VIII,41-49
doi: https://doi.org/10.26562/ijirae.20210.v0803.003
Editor-Chief: Dr.A.Arul Lawrence Selvakumar, Chief Editor, IJIRAE, AM Publications, India

Abstract: Soil is a critical part of successful agriculture and is the source of the nutrients that we use to grow crops. There are different types of soil and there are different properties of each soil. On these different properties, several types of crops grow. We need to know the properties and characteristics of various soil types to understand which crops sow in certain soil types. Machine Learning allows the user to feed a computer algorithm on an immense amount of data and have the computer analyze, make data-driven recommendations and decisions based to analyze the input data. Machine Learning techniques are used to model this process. Machine Learning has come into the picture with the big data technologies and high-performance computing that create new opportunities for data-intensive science in the multi-disciplinary agri-technology domain. In this paper, we have proposed a model that can find whether the soil is fertile or not, Sowing crop seed on fertile soil, and at last predicting the crop yield on different soil features. According to prediction, it can be suggested and recommended which crops grow more. Various Machine Learning algorithms such as Support Vector Machine (SVM), Random Forest, Naive Bayes, Linear Regression, Multilayer perceptron (MLP), and ANN are used for soil classification and crop yield. Test results show that the proposed ANN method follows a deep learning architecture which means it has several layers for input and output are connected to achieve better accuracy than numerous existing methods.

Keywords: Fertility, Crop Yield Prediction, Soil, Chemical Features, ANN, Machine Learning.

I. INTRODUCTION

Agriculture is the technique of cultivating the soil, growing crops, and raising livestock. It builds the preparation of plant and animal products for people to use and their distribution to markets. Agriculture gives most of the nation’s food and fabrics. Agriculture plays a very important role in the global economy. The world population is increasing at a very fast rate and with an increase in population, the need for food is also increasing. Traditional methods used by farmers are not sufficient enough to serve the huge demand and so they have to hamper the soil by using harmful pesticides in an intensified manner. This affects the agriculture practice a lot and in the end, the land remains barren with no fertility.

Machine Learning in agriculture is used to improve the product quality of the crops in the agriculture sector. Machine Learning is the scientific field which gives the machine the ability to learn without being intervention of human being. The main aim of this research is to design the crop yield prediction and soil fertility analysis model by machine learning supervised and deep neural network model. The real-time data of soil and crop are collected from the different online repository (Private). To assessed the model these datasets are used and computed the result. The datasets are divided into two category training dataset and testing dataset to build the model. Then different Machine learning algorithms are applied to classify the soil whether the soil is fertile or not by using soil micro-nutrients and chemical features. The crop yield, crop sowing is also analyzed using this ML technique. The algorithm assessed involves SVM, artificial neural network (ANN), decision tree, Naive Bayes, and linear regression which all are available in the machine learning model.

II. LITERATURE REVIEW

Here we take a few of the papers related to Soil analysis and Crop yield prediction using various machine learning techniques and some of them showdown. Machine Learning is a decision making to solve a problem that needs to be modeled from data. In Paper [1] author used the dataset, collected from Soil Resources Development Institute (SRDI). They used KNN, Bagged Trees, SVM. The dataset consists of 495 samples of 11 classes. A model is designed for predicting soil series and giving suitable crop yield suggestions for that specific soil.

Here among all the classification SVM has given the highest accuracy in Soil Classification. Pramudyana Agus, Noor Akhmad, and Teghu Bharata [2] used several machine learning algorithms, such as neural networks, decision trees, naive Bayes, and SVM. The algorithm used to automate soil type classification with satisfactory accuracy (> 70%). Mahesh Gauda Patil and Indira R. Umanji [3] worked on crop protection in organic agriculture. They used deep learning models that were developed, based on specific Convolutional neural network architectures. The research is developed for checking the various crop diseases to help the farmer. Ashwini Rao, Janhavi U, Abhishek Gowda NS and Manjunath [4] describes an approach for classifying and grading the soil samples using different scientific features. Different algorithms are used to extract different features of soil like color, texture, etc commercial imaging libraries with Digital Signal Processing (DSP) boards are implemented for real-time operations. It involves both image processing and pattern recognition techniques.

Sini Anna Alex and Anita Kanavalli [5] in this paper, precision agriculture is implemented for various crop yield prediction using a convolutional neural network algorithm. Here author's focus is on optimizing the significant parameters such as rainfall, temperature, and fertilizer rate to obtain the P-values for testing the crop. Sikha Prakash, Animesh Sharma, and Sitansu Shekhar [6] introduced machine learning techniques for the prediction of soil moisture in advance. They used different ML algorithms viz multiple linear regression, support vector regression, and recurrent neural networks for the prediction of soil. These techniques were applied to their different databases collected from different online repositories. Their performance of the prediction is evaluated based on mean squared error (MSE) and coefficient of determination (R^2). The comparison result shows that multiple regression is Superior providing MSE and R^2 of 0.14 and 0.975. Jay Gholap et al. [7] used soil datasets from three regions (Khed, Bhor, and Velhe) of the Pune district, India.

Dataset has a total of 1988 instances with 9 attributes. They focus on applying various algorithms such as Naive Bayes, JRip, J48 (which is an open-source Java implementation of the C4.5 decision tree algorithm) for the classification task. Madhavi Gudavalli and Vidyasree [8] describes clustering techniques. This paper represents a study on different clustering techniques that are incorporated on the seed data sets to enhance the clustering approach based on the various parameter like area, perimeter, compactness, length, a width of the kernel asymmetric coefficient, and length of the kernel groove.

Methodology:-

A. Dataset: - There are three different datasets are used for the prediction of the model.

i) Soil Dataset
ii) Crop Dataset
iii) Yield Dataset

i) Soil Dataset: It consists of 15 attributes like PH, EC, OC, OM, N, P, K, Zn, Fe, Cu, Mn, Sand, Silt, Clay, CaCo3, and CEC. From this all attributes we classified and analyzed by applying ML model whether the Soil is fertile or not.

ii) Crop Dataset: It consists of 4 attributes like temperature, humidity, PH, rainfall. Crop Prediction is performed by using a different algorithm.

iii) Yield Dataset: It consists of 6 attributes are Nitrogen (N), Phosphorous (P), Potassium (K), Organic Care (Og), PH, temp. Yield Prediction is performed on these different attributes using an ML algorithm.

B. Method and Experimentation: - Datasets are collected from different resources then classified the data and group into two sets: -

i) Training Dataset,
ii) Testing Dataset

a) Implementation using ML algorithm : Different Supervised algorithm and compared the result and accuracy with the models.

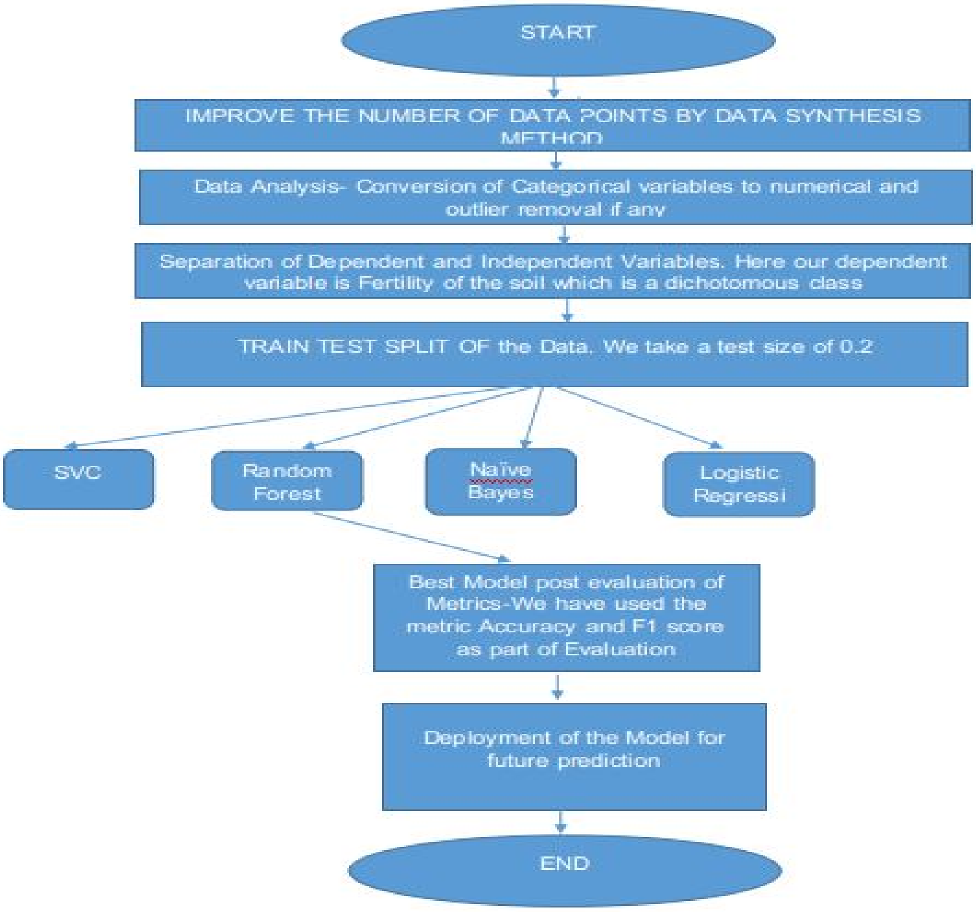

START
IMPROVE THE NUMBER OF DATA POINTS BY DATA SYNTHESIS METHOD
Data Analysis- Conversion of Categorical variables to numerical and outlier removal if any
Separation of Dependent and Independent Variables. Here our dependent variable is Fertility of the soil which is a dichotomous class
TRAIN TEST SPLIT OF the Data. We take a test size of 0.2
SVC
Random Forest
Naïve Bayes
Logistic Regressi
Best Model post evaluation of Metrics-We have used the metric Accuracy and F1 score as part of Evaluation
Deployment of the Model for future prediction
END

Fig.1.1 Flow diagram to predict if the soil is fertile or Not

1. Classifying if the soil is Fertile or not:

i. In this model, we take various attributes of the soil such as Ph, EC, OC, OM, N, P, K, etc. to determine if the soil is fertile or not.
ii. To classify, we have first taken SVC and the accuracy has come around 0.82.
iii. Next, we tried Multinomial Naïve Baye's model and the accuracy has come around 0.815.
iv. Finally, we tried Random Forest and the accuracy came up to 1 with an F-1 score of 1. This is no ordinary accomplishment, because we may get high Accuracy for imbalanced datasets, but in this case, the F-1 score along with accuracy is 1 implying the best possible model to classify if the soil is fertile or not.

2. Predicting the type of Crop to grow:

i. We try to predict the type of crop-based on attributes like temp, pH, humidity, rainfall.
ii. Here we use a multi-class classification model to predict the type of crop to be grown.
iii. First, we adopt SVC. By using this model we could accomplish only 63.870% accuracy.
iv. The next model we chose to predict the crop label is MLP Classifier. It was also not that great as we could accomplish the accuracy of 71.9.
v. The next model we used is the Multinomial NB model. It was also pretty bad as we could accomplish only 54% accuracy.
vi. We then used Random Forests, with an accuracy of 94%, it was the best model so far. So we chose Random Forest Classifier as the perfect model to choose the type of crop to be grown given its soil and weather attributes.

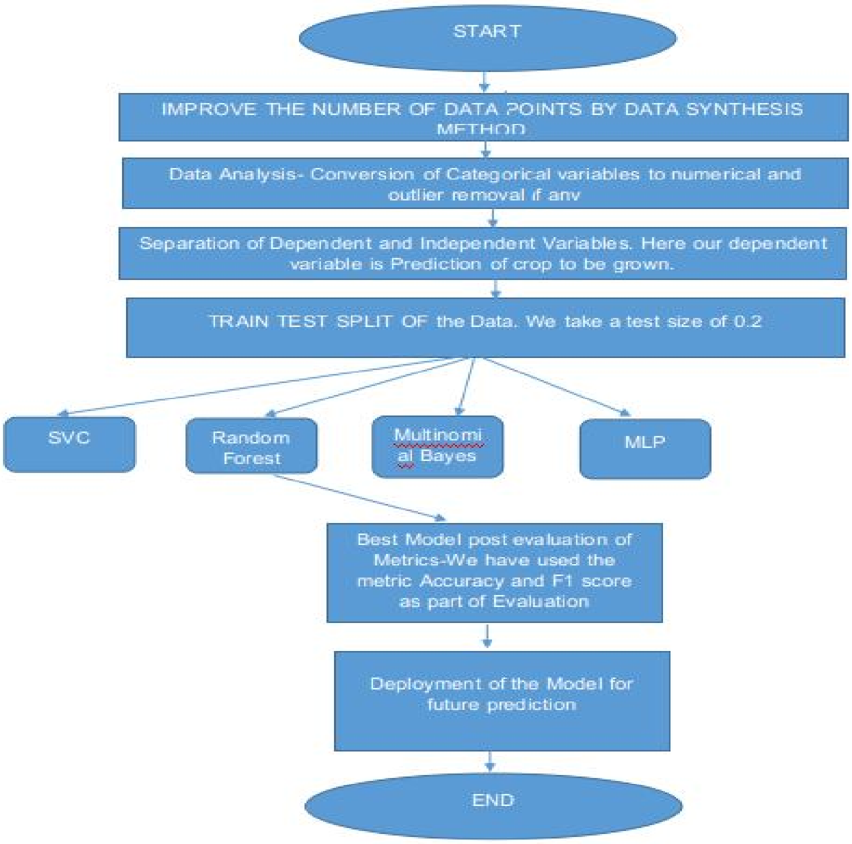

START
IMPROVE THE NUMBER OF DATA POINTS BY DATA SYNTHESIS METHOD
Data Analysis- Conversion of Categorical variables to numerical and outlier removal if any
Separation of Dependent and Independent Variables. Here our dependent variable is Prediction of crop to be grown.
TRAIN TEST SPLIT OF the Data. We take a test size of 0.2
SVC
Random Forest
Multinomial Bayes
MLP
Best Model post evaluation of Metrics-We have used the metric Accuracy and F1 score as part of Evaluation
Deployment of the Model for future prediction
END

Fig 1.2 Flow diagram to predict the type of crop to be grown

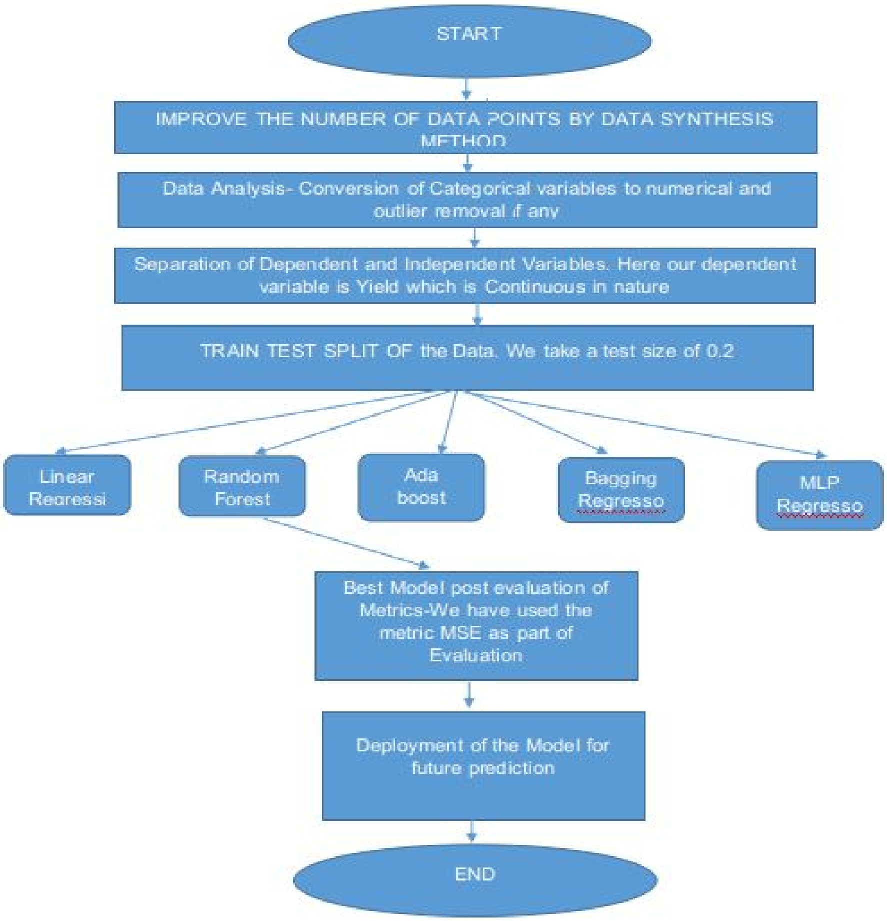

START
IMPROVE THE NUMBER OF DATA POINTS BY DATA SYNTHESIS METHOD
Data Analysis- Conversion of Categorical variables to numerical and outlier removal if any
Separation of Dependent and Independent Variables. Here our dependent variable is Yield which is Continuous in nature
TRAIN TEST SPLIT OF the Data. We take a test size of 0.2
Linear Regressi
Random Forest
Ada boost
Bagging Regresso
MLP Regresso
Best Model post evaluation of Metrics-We have used the metric MSE as part of Evaluation
Deployment of the Model for future prediction
END

FIG 1.3 Flow Diagrams to Predict the Yield of a Crop

3. Predicting the Yield of a crop:

i. To Predict the yield of a crop, we have taken a 'yield dataset' with various attributes such as Nitrogen, Phosphorous, Potassium, Organic carbon, pH, and temperature determining the yield of a crop. As the yield of a crop is continuous we adopt regression techniques to predict the yield of the crop. We would consider various models to perform the regression technique and evaluate each model by finding the mean square error for each model. The best model is the one that has the least mean square error. The first model that we have performed is Linear Regression. It was not ideal. The MSE was around 190845.

ii. The second model is the Random Forest model. It was excellent as it has the minimum mean squared error with 27392.

iii. We then considered Adaboost Regressor to check if we could achieve lower MSE, but we managed to get an MSE of 195044

iv. We then tried Bagging Regressor. It was good but not better than Random Forest Regressor as it managed to achieve an MSE of 30828.

v. We also tried MLRegressor and SVR, but they couldn't match the MSE of Random Forest Regressor. Thus Random Forest Regressor is the best model to predict the yield of the crops.

Random Forest:

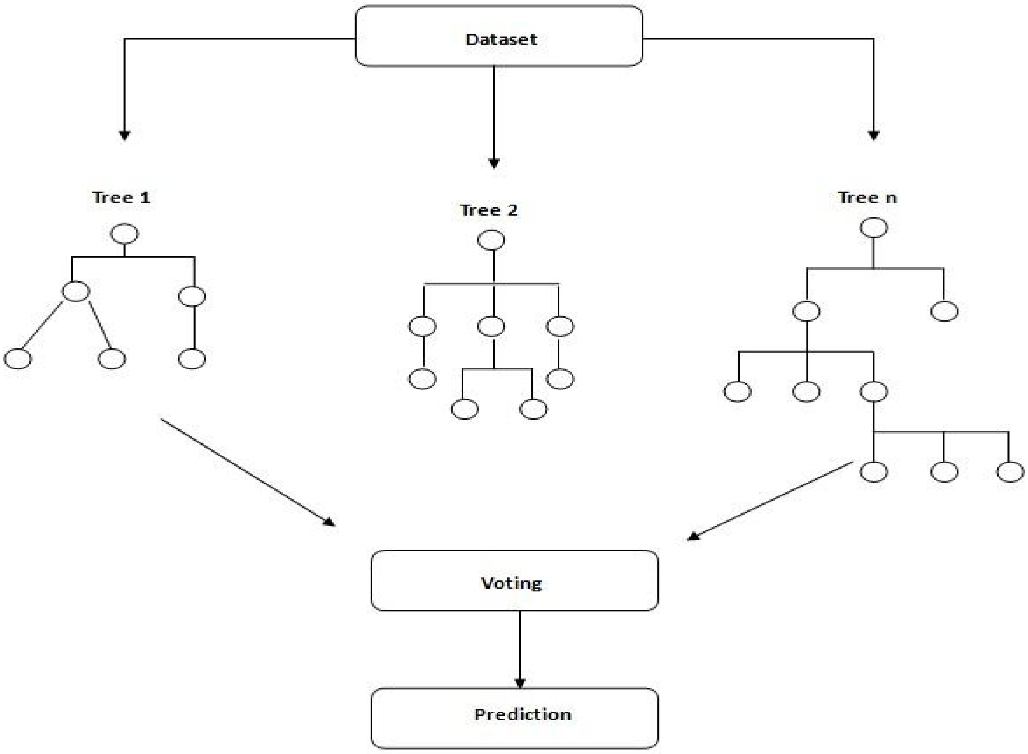


Figure: 1.4 Random Forest

Random forest is a supervised machine learning algorithm. It has random decision forests with an ensemble learning method for classification problems; it has usually trained with the “bagging” method. The bagging method is that a mixture of learning models increases the overall result. It uses for both classification and regression problems.

Random forest works on two main steps:

a) Creation

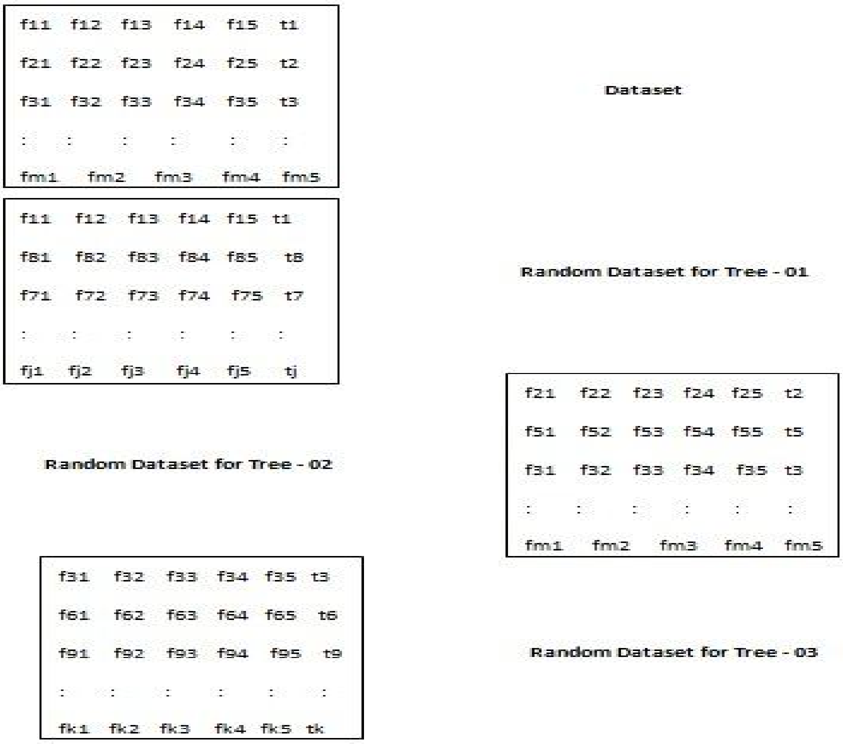
b) To make a prediction

a) Creation: Pseudo code

i) It select ‘k’ features from total ‘m’ features randomly where k<<m.
ii) Among the ‘k’ features, calculate node ‘d’ using the best split point.
iii) Split the node into child nodes using the best split.
iv) Repeat the a to c steps until ‘l’ number of nodes has been reached.
v) Builds forest by repeating step a to d for ‘n’ number times to create ‘n’ number of trees.

b) Prediction:

i) Take the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target).
ii) Calculate the high voted predicted target as the final prediction from the random forest algorithm.



f11	f12	f13	f14	f15	t1
f21	f22	f23	f24	f25	t2
f31	f32	f33	f34	f35	t3
:	:	:	:	:	:
fm1	fm2	fm3	fm4	fm5	

Dataset

f11	f12	f13	f14	f15	t1
f81	f82	f83	f84	f85	t8
f71	f72	f73	f74	f75	t7
:	:	:	:	:	:
fj1	fj2	fj3	fj4	fj5	tj

Random Dataset for Tree - 01

f21	f22	f23	f24	f25	t2
f51	f52	f53	f54	f55	t5
f31	f32	f33	f34	f35	t3
:	:	:	:	:	:
fm1	fm2	fm3	fm4	fm5	

Random Dataset for Tree - 02

f31	f32	f33	f34	f35	t3
f61	f62	f63	f64	f65	t6
f91	f92	f93	f94	f95	t9
:	:	:	:	:	:
fk1	fk2	fk3	fk4	fk5	tk

Random Dataset for Tree - 03

Figure: 1.5 Random Selection features

Result:

The proposed method is based on these three dataset described above. Several machine learning methods are applied separately to recognize the soil Fertility of test sample, different crop of test sample and crop yield test sample. In result analysis model, we have calculated precision, recall, f1-score and support with accuracy and macro avg and weighted avg result. The classification report and confusion matrix for each algorithm is also obtained from result analysis model. Neural Network and Random Forest classifier have better class recall. SVM and Naive Bayes the lowest class recall. Except for Random forest algorithm, all algorithms have several class precision below 55%. The summary of accuracy using all attributes is shown at figures 2.0.

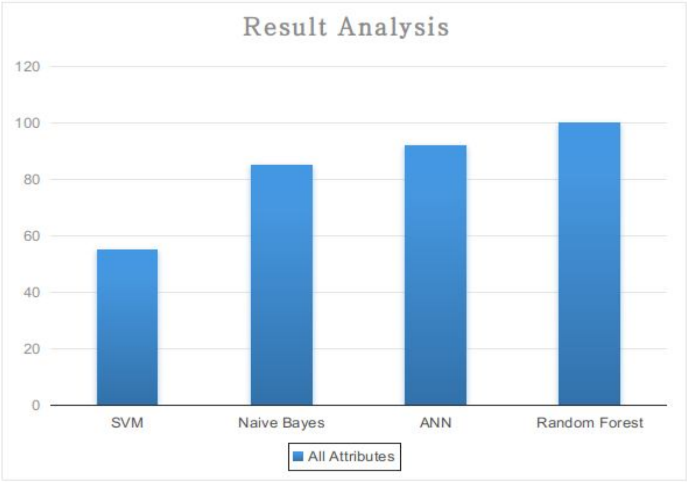


Fig. 2.0 Showing accuracy for all attributes

The accuracy of all methods proposed in this paper (SVM, neural network, decision tree, naive bayes, Random forest, Artificial Neural Network, Regression model) is better than previous experiment by SK AL Zaminur and Kaushik (KNN, Bagged Trees and Gaussian kernel based Support Vector Machines(SVM)). In this paper, we have applied over sampling method to improve synthetic data and re-sampled the data. Imported imblearn package for Random Over Sampler, with this all the our algorithms achieve comparative accuracy, but random forest shows better accuracy than other method used here. As table 2.1 shows below the classification accuracy.

Table 2. Result of the Proposed Method

	Method	Accuracy(%)
Gholap, Jay, et al [7]	J48	92.3
SK, Kaushik, S.M [1]	SVM	94.95
Proposed Work	Random Forest	100

Table: 3. Show regression metrics: For yield prediction model with MAE,MSE and R^2.

Model	Mean Absolute Error	Mean Squared Error	R^2 (R square)
Linear Regression	389.679	199735.585	0.034
Random Forest	59.387	15819.658	0.916
Adaptive Boost	382.250	193993.485	0.007
Bagging Regressor	65.975	20883.361	0.893
Gradient Boosting	312.074	138409.677	0.281
Neural Network	391.545	209409.226	0.086
SVM	388.463	191805.504	0.035

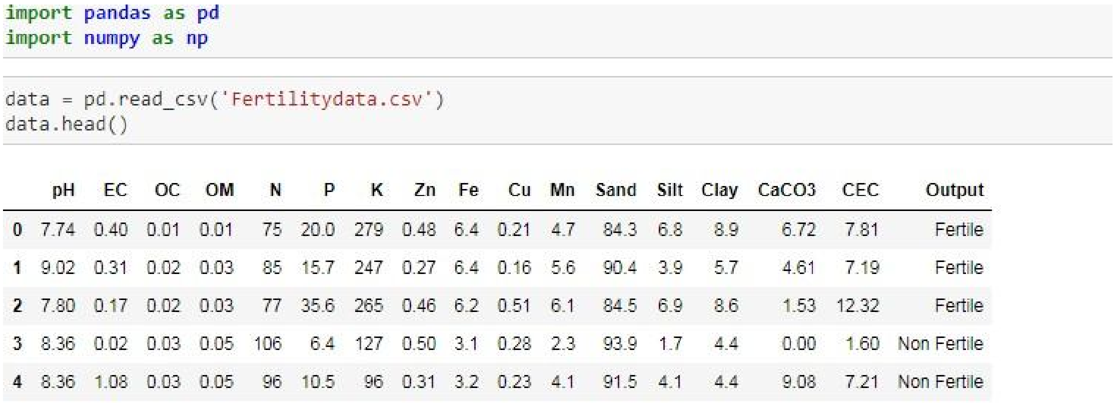
TABLE 3. Regression Metrics

Here in the table 2.2 show metrics result that Random Forest has better accuracy compared with Linear Regression, Ada Boost, Bagging Tree, Gradient Boosting, Neural Network and SVM.

b) Implementation using ANN : Applied Artificial Neural Network using Keras Library.

Steps: There are 6 step are present

Step 1: Load Data

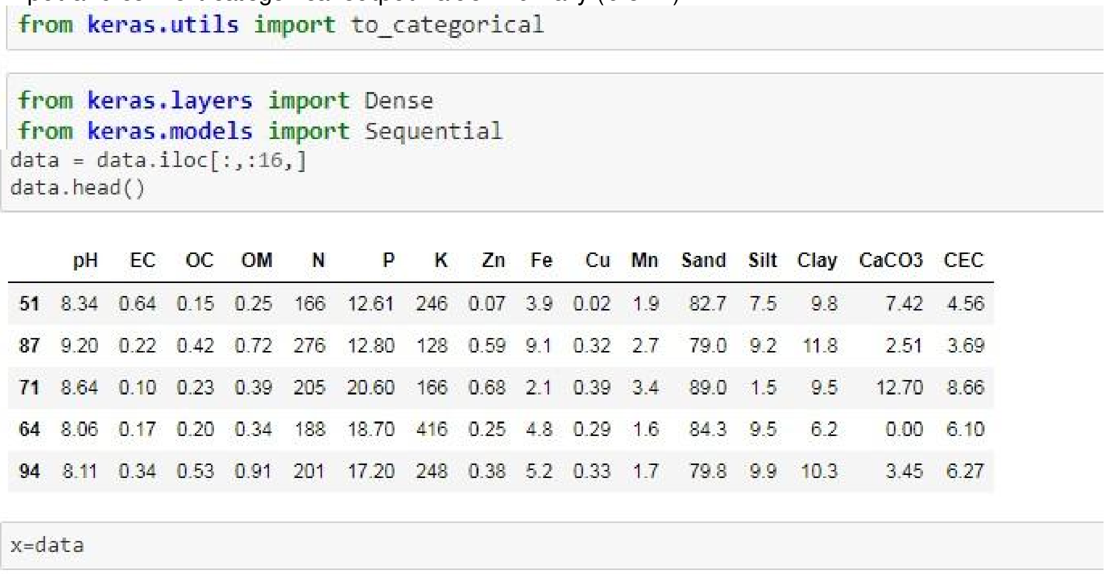


```
import pandas as pd
import numpy as np
```

```
data = pd.read_csv('Fertilitydata.csv')
data.head()
```

	pH	EC	OC	OM	N	P	K	Zn	Fe	Cu	Mn	Sand	Silt	Clay	CaCO3	CEC	Output
0	7.74	0.40	0.01	0.01	75	20.0	279	0.48	6.4	0.21	4.7	84.3	6.8	8.9	6.72	7.81	Fertile
1	9.02	0.31	0.02	0.03	85	15.7	247	0.27	6.4	0.16	5.6	90.4	3.9	5.7	4.61	7.19	Fertile
2	7.80	0.17	0.02	0.03	77	35.6	265	0.46	6.2	0.51	6.1	84.5	6.9	8.6	1.53	12.32	Fertile
3	8.36	0.02	0.03	0.05	106	6.4	127	0.50	3.1	0.28	2.3	93.9	1.7	4.4	0.00	1.60	Non Fertile
4	8.36	1.08	0.03	0.05	96	10.5	96	0.31	3.2	0.23	4.1	91.5	4.1	4.4	9.08	7.21	Non Fertile

We use the Numpy library to load our dataset and use Keras library to define our model. Indian Soil fertility dataset we loaded to test the model. It describes soil chemicals features with output value whether the soil is fertile or not. It is a categorical classification problem (1 or 0), we converted output value fertile as 0 and not fertile as 1. The entire input variable that describes soil features is numerical. It easy to use input variables with a neural network that expect numerical input and convert categorical output value in binary (0 or 1).



```
from keras.utils import to_categorical
```

```
from keras.layers import Dense
from keras.models import Sequential
data = data.iloc[:,:16,]
data.head()
```

	pH	EC	OC	OM	N	P	K	Zn	Fe	Cu	Mn	Sand	Silt	Clay	CaCO3	CEC
51	8.34	0.64	0.15	0.25	166	12.61	246	0.07	3.9	0.02	1.9	82.7	7.5	9.8	7.42	4.56
87	9.20	0.22	0.42	0.72	276	12.80	128	0.59	9.1	0.32	2.7	79.0	9.2	11.8	2.51	3.69
71	8.64	0.10	0.23	0.39	205	20.60	166	0.68	2.1	0.39	3.4	89.0	1.5	9.5	12.70	8.66
64	8.06	0.17	0.20	0.34	188	18.70	416	0.25	4.8	0.29	1.6	84.3	9.5	6.2	0.00	6.10
94	8.11	0.34	0.53	0.91	201	17.20	248	0.38	5.2	0.33	1.7	79.8	9.9	10.3	3.45	6.27

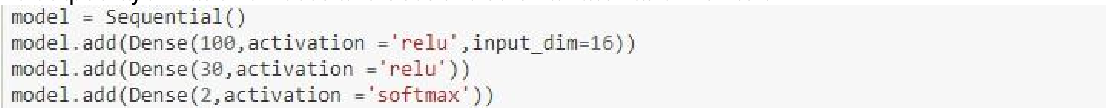
```
x=data
```

Step 2: Define model

Models in Keras have a sequence of layers. We create a Sequential model and added layers with our network architecture. First Layer it has input dim argument and setting it 16 for the 16 input variables. We defined a connected network structure with 3 layers. Connected layers are defined using the dense class. We specified the number of neurons/nodes in the layer as the first argument, and activation function using the activation argument.

We used the rectified linear unit (RELU) activation function on the first two layers and the softmax function in the output layer. To achieve better performance Relu activation function is used. We used a softmax on the output layer to ensure our network output is between 0 and 1 as Yes and No.

- The model has 16 variable rows of data(the input_dim=16 argument)
- The first hidden layer has 100 nodes and uses the relu activation function.
- The second hidden layer has 30 nodes and uses the relu activation function.
- The output layer has two nodes and uses the softmax activation function.

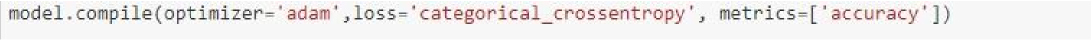


```
model = Sequential()
model.add(Dense(100,activation ='relu',input_dim=16))
model.add(Dense(30,activation ='relu'))
model.add(Dense(2,activation ='softmax'))
```

Step 3: Compile model

Now the model has defined, we compiled it in this step.

Model compilation uses efficient numerical libraries such as Theano or TensorFlow. The backend has automatically chosen the best way to represent the network for training and making predictions to run on our hardware, such as CPU or GPU or even distributed. While compiling, we specified some additional properties required when training the network. Trained the network means found the best set of weights to map inputs to output in our dataset. We also specified the loss function to evaluate a set of weights, the optimizer is used to search through different weights for the network and any optional metrics we would like to collect and report during training. In this model, we used cross-entropy as the loss argument. This loss is shown for categorical classification problems and is defined in Keras as "categorical_crossentropy". We defined the optimizer as the efficient stochastic gradient descent algorithm "adam". It is a popular version of gradient descent because it automatically tunes itself and gives good results in a wide range of problems. Hence it is a classification problem; we collected and reported the classification accuracy, defined via the metrics argument.



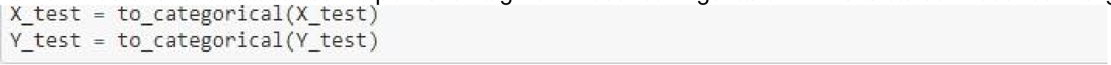
```
model.compile(optimizer='adam',loss='categorical_crossentropy', metrics=['accuracy'])
```

Step 4: Fit model

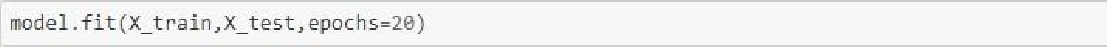
After compiled the model next phase is efficient computation. The model executed on some data. Train and fit our model on our loaded data by calling the fit() function on the model. Training occurs over epochs and each epoch is split into batches.

Epoch: One undergoes all of the rows within the training dataset.

Batch: One or more samples considered by the model within an epoch before weights are updated. One epoch is comprised of one or more batches, based on the chosen batch size and the model is fit for many epochs. The training process ran for a fixed number of iterations through the dataset called epochs. We also set the numbers of dataset rows that are considered before the model weights are updated within each epoch as called the batch size and set using batch_size argument. In this model, we ran for a number of epochs (20) and use a relative default batch size of 32. These configurations can be chosen experimentally by trial and error. We trained the model enough so that it learned a good mapping of rows of input data to the output classification. The model will always have some error, but the amount of error would level out after some point for a given model configuration. This is called model convergence.



```
X_test = to_categorical(X_test)
Y_test = to_categorical(Y_test)
```

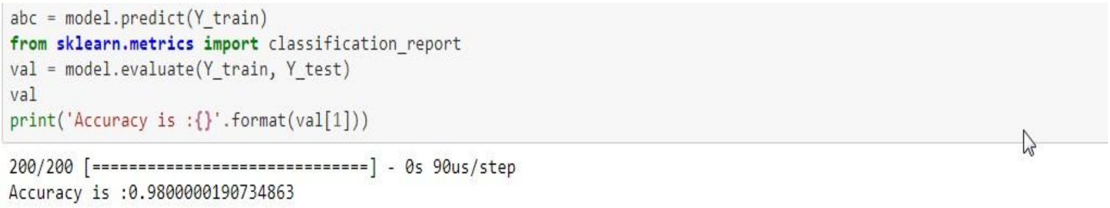


```
model.fit(X_train,X_test,epochs=20)
```

It required CPU or GPU for performing the execution of large models.

Step 5: Evaluate

Now we have trained our neural network on the entire dataset and we can evaluate the performance of the network on the same dataset. We also trained and tested the dataset for training and evaluation of our model. Evaluated the model on training and test dataset using evaluate() function on the model and passed the input and output. This has generated a prediction for each input and output paired and collected scores, including the average loss and any metrics which has configured, such as accuracy. This evaluate() function returns a list with two values. The first will be the loss of the model on the dataset and the second will be the accuracy of the model on the dataset. We are only interested in reporting the accuracy, so we ignored the loss value.



```
abc = model.predict(Y_train)
from sklearn.metrics import classification_report
val = model.evaluate(Y_train, Y_test)
val
print('Accuracy is :{}'.format(val[1]))

200/200 [==============================] - 0s 90us/step
Accuracy is :0.9800000190734863
```

We achieved a very good accuracy of 98% using neural network.

Step 6: Make Predictions

For the prediction, we used the predict() function on the model. The value which we have predicted for the above model, the result is 0 and 1.

CONCLUSION:

A model is proposed for predicting the soil fertility and crop yield with types of the crop can grow on fertile soil. The research has done on soil datasets and crop datasets of the Indian region. The model has been tested by using a different machine learning algorithm. Multi-Layer Perceptron and Random Forest Classifier show good accuracy among all the classifiers but ANN has given the highest accuracy in soil fertility prediction, crop prediction, and crop yield prediction. ANN works on input dimensions that are fed and input is provided. Input nodes are connected to the first layer and the corresponding sum of product is performed by adding the bias and weight terms. This output is then passed through a filter and passed as an input to the second hidden layer and the sequence goes till one gets the output. The Model has to predict either 1 or 0 the sigmoid function filter is placed in the last hidden layer. The output is then compared against the actual value and the feedback mechanism is involved and the weights are updated. This process is done until all the weights are set and finally produce the minimal value of the difference of predicted and actual values. This completes one Epoch. Multiple epochs are run until the accuracy reaches the optimal value. Thus the accuracy and F1 score would be really high for a model developed from Keras. In the future, providing fertilizer usage for crop production is our goal and also data from another region will be added to make this model more reliable and efficient usage.

REFERENCES

1. SK AL Zaminur Rahman, Kaushik Chandra Mitra and S.M Mohidul Islam, "Soil Classification using Machine Learning Methods and Crop Suggestion Based on Soil Series", 2018 IEEE, 21st International Conference of Computer and Information Technology (ICCIT), pp.978-1-5386-9242-4/18
2. Pramudyana Agus Harlianto, Teguh Bharata Adji and Noor Akhmad Setiawan, "Comparison of Machine Learning Algorithm for Soil Type Classification", 2017 3rd International Conference on Science and Technology-Computer(ICST).
3. Maheshgouda Patil, Indira R. Umarji, "Identification of Crop Diseases using Deep Learning", International Journal of Research in Engineering,Science and Management, Vol-2,Issues-6,June-2019, ISSN: 2581-5792
4. Ashwini Rao, Janhavi U, Abhishek Gowda NS, Manjunatha and Mrs.Rafega Beham A, "Machine Learning in Soil Classification and Crop Detection", IJSRD- International Journal for Scientific Research and Development, Vol-4, Issue 01, 2016, ISSN: 2321-0613
5. Sini Anna Alex and Anita Kanavalli, "Intelligent Computational Techniques for Crops Yield Prediction and Fertilizer Management over Big Data Environment", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Vol-8, Issue-12, October 2019, ISSN: 2278-3075
6. Shikha Prakash, Animesh Sharma and Sitanshu Shekhar Sahu, "Soil Moisture prediction using Machine Learning", 2018 IEEE, 2nd International Conference on Inventive Communication and Computational Technologies(ICICCT)
7. Jay Gholap, Anurag Ingole, Jayesh Gohil, Shailesh Gargade and Vahida Attar, "Soil Data Analysis Using Classification Techniques and Soil Attribute Prediction", arXiv preprint arXiv:1206.1557.
8. Dr Madhavi Gudavalli, Vidyasree P and S Viswanadha Raju, "Clustering Analysis for Appropriate Crop Prediction using Hierarchical, Fuzzy C-Means, K-Means and Model based Techniques", Vol-4, Issue 11, November- 2017, Scientific Journal of Impact Factor (SJIF): 4.72, ISSN (p): 2348-6406
9. Neural Network Algorithm, https://www.investopedia.com/terms/n/neuralnetwork.asp
10. Support Vector Machine (SVM) Algorithm, https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/