WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Implementing Machine Learning Models in Predicting Earthquake

Dinky Tulsi Nandwani[1], Vanita Buradkar[2]

[1]Rajiv Gandhi College of Engineering Research & Technology, Chandrapur, Maharashtra, India
Ghugus, Maharashtra, India
dinky.nandwani014@gmail.com
[2]Rajiv Gandhi College of Engineering Research & Technology, Chandrapur, Maharashtra, India
Chandrapur, Maharashtra, India
vsburadkar@gmail.com

ABSTRACT:

This research paper explains the feature selection and model building. After preprocessing of data, feature selection is done. A data set contains numerous of features which are random and may not be useful in prediction. Feature Selection deals with reduction of random features under consideration and obtaining a set of minimum features which contribute to accurate prediction. The accuracy of the models is decreased if it contains irrelevant features. It is the most important concept of machine learning that affect the performance of the model. Random forest and xgboost algorithm are used for feature selection and neural network is also used for model building in project.

Keywords: Feature Selection, Model Building, Machine Learning, Random Forest, Xgboost, Neural Network.

1. INTRODUCTION

As the era is evolving and assisting humans for a higher and a convenient way of life, possibility at saving existence is taken up with the help of green ML algorithm and records technological know-how to provide accurate forecast. Gadget getting to know is a subset of artificial Intelligence. It permits the device to evolve to a behavior of a particular kind based on its very own learning and possesses the ability to improve itself certainly completely from experience without any express programming, human mediation or help [1]. Initialization of a device gaining knowledge of technique begins with feeding a sincere pleasant facts-set to the set of rules(s), on the way to construct a ML prediction version. Algorithms carry out information discovery and statistical evaluation, figuring out styles and developments in data. Choice of algorithms is predicated on facts and on the mission that requires automation.
ML algorithms assemble two styles of predictive fashions, Regression and type fashions [2]. Every of this techniques facts in a unique manner. The dataset of this challenge is labeled correctly, so its miles part of supervised learning. Class is a category of supervised learning.

2. LITERATURE REVIEW

There are numerous gadgets getting to know algorithms, and every set of rules has its personal blessings and drawbacks for solving geosciences issues. On this paper, we made a contrast evaluation of 5 algorithms: k-nearest acquaintances (kNN) [3], the decision Tree [4], the Random forest Classifier (RFC) [5], and the acute gradient boosting (XGBoost) [6], LightGBM [7]. On this research, we used "scikit-examine" [8], the developed python framework for utilization of kNN, Desicion Tree, and Random woodland classifier. XGBoost and LightGBM have their very own python framework.

Ok-Nearest pals (kNN) are a system getting to know approach that has been used for data mining [3]. Each factor (statistics point) has location in a multidimensional area, where the distance includes axis or functions of modern datasets. The skilled version defines a most suitable matter of friends for the educated dataset and while we have a brand new (test) facts factor the version reveals the ok nearest pals for the take a look at dataset. KNN has the advantage of being nonparametric.

Decision tree methods are data mining techniques, and that they have been efficiently used for type troubles. Selection trees have been advanced with the aid of Morgan and Sonquist in 1963, and that they applied the set of rules for determinants of social situations [4]. One advantage of the decision timber is that they're computationally rapid and can cope with high-dimensional facts. On the other hand, an unmarried decision tree can overfit at the information and the set of rules is grasping; therefore, it maintains growing deeper in the tree. The

random woodland changed into added through Breiman as a gaining knowledge of tree classifier of an ensemble [5]. The important thing concept of the set of rules is to take the values of a random vector from an aggregated bootstrap pattern (teach dataset) after which to educate many decision bushes. However, the trained tree could have a number of trees, thus it calls for extra computational assets.
The primary advantage of the XGBoost is parallelization. XGBoost is a scalable version of the gradient boosting gadget algorithm and confirmed efficiency in several system mastering packages. In [6], the XGBoost is an ensemble of classification and regression trees and works for records with nonlinear features. The key concept is to use vulnerable trees and enhancement of trees accuracy for every new release, taking account the mistake in prediction from the previous end result of a vulnerable tree, the next tree classifier is trained to don't forget the error of the already trained ensemble.

LightGBM is a quite new framework and has a huge utility in system gaining knowledge of/records science applications. The primary problem of gradient boosting algorithms is that the algorithm processes all records to advantage the result of feasible separation factors, which impacts performance. This technique has been changed to enhance the highest quality seek approach [7].

3. FEATURE SELECTION

It entails both function selection or characteristic Extraction and function Scaling. A statistics set includes numerous of capabilities which can be random and may not be beneficial in prediction. Characteristic Engineering deals with discount of random capabilities below attention and acquiring a hard and fast of minimum features which contribute to correct prediction. Many algorithms are furnished via ML for function selection/extraction. Function scaling is strategy used to standardize or normalize the variety of capabilities within the information-set. Feature Engineering is useful as it compresses the records, reduces the garage space, computation time and gets rid of redundant functions.

4. MODELLING

The yield of an ML set of rules is a 'version'. To start with, the target variable and function variable are comprehended and fetched. 2nd, the statistics-set is partitioned into schooling and checking out records-set and 0.33; the regressor/classifier version is constructed and suited to schooling data-set. In python, scikit-analyze is a easy, simple, green open supply library that executes a variety of device getting to know algorithms presenting numerous category, regression and clustering algorithms the usage of a unified interface.[9] little by little constructing is as follows:

4.1. Building a Random Forest Model:

Random forests are an ensemble learning technique that may be fabricated for each regression in addition to category chore. It takes on the challenge of building multiple of choice bushes in the course of schooling and outputs the elegance this is mean prediction (regression) of each person tree or the mode of the training (classification). This huge number of timber represents a forest. Choice bushes are rule based totally fashions; on a given education information-set with targets and capabilities, the selection tree algorithm will give you regulations to perform classification and regression. Features could be nodes and their presence and shortage will represent likeliness. This helps in constructing a direction of rules to paintings with. The basis and splitting node is based totally on data benefit or gini index [10]. In Random wooded area, the basis and splitting nodes are calculated in a random way.

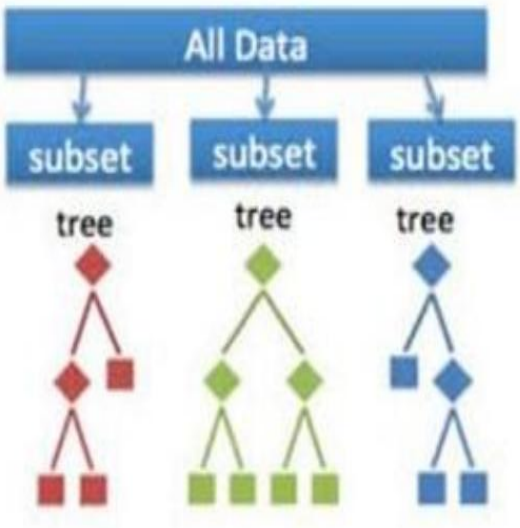

Fig 1: Basic diagram of Random forest model

The output of the random forest algorithm is shown as:

```
In [17]:
rfClassifier = RandomForestClassifier()
model = rfClassifier.fit(X_train_os, y_train_os_one_hot)

/opt/conda/lib/python3.6/site-packages/sklearn/ensemble/forest.py:246: FutureWarning: The default value of n_estimators will c
hange from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)

In [18]:
dtClassifier = DecisionTreeClassifier(max_depth=100, min_samples_leaf=10, class_weight='balanced')
dtClassifier = DecisionTreeClassifier()
model = dtClassifier.fit(X_train_os, y_train_os_one_hot)

In [19]:
model

Out[19]:
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=None,
            splitter='best')
```


Fig 2: Code of Random forest algorithm

```
In [20]:
y_train_pred = model.predict(X_train_os)
Accuracy = accuracy_score(y_train_os_one_hot, y_train_pred) * 100
F1_score = f1_score(y_train_os_one_hot, y_train_pred, average='weighted')

In [21]:
y_train_pred

Out[21]:
array([[0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 1.],
       [1., 0., 0., 0., 0.],
       ...,
       [1., 0., 0., 0., 0.],
       [1., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1.]])

In [22]:
Accuracy, F1_score

Out[22]:
(99.76398986291451, 0.9987727872553426)
```

Fig 3: Accuracy prediction of Random forest model

```
feature_importance_df = feature_importance(model, X_train_os)
feature_importance_df.iloc[:35].plot('bar')
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f75fa2549e8>
```

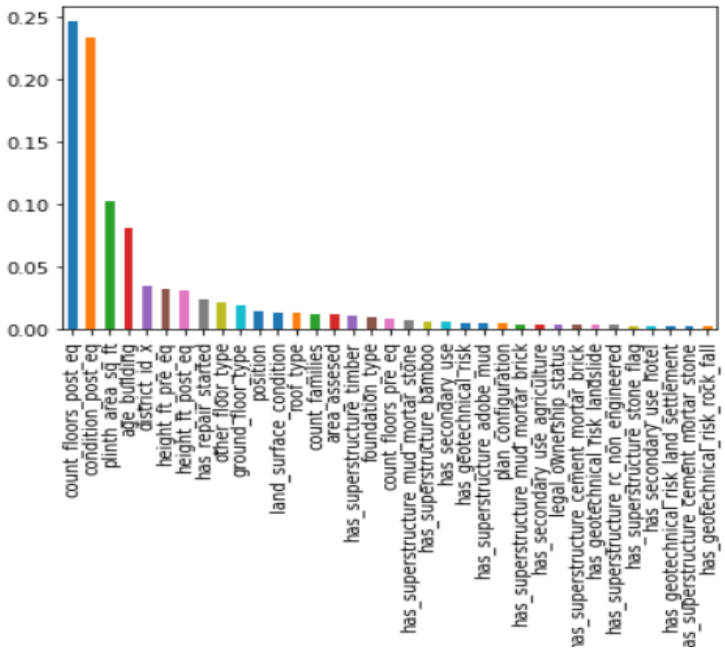

Fig 4: Feature importance of attributes using Random forest model

Table 1: Grade prediction using Random Forest algorithm

1	building_id	damage_grade
2	a3380c4f75	Grade 4
3	a338a4e653	Grade 5
4	a338a4e6b7	Grade 5
5	a33a6eaa3a	Grade 3
6	a33b073ff6	Grade 5
7	6604e4896c6	Grade 2
8	a33b07430f	Grade 3
9	a33c386cf3	Grade 5
10	a33c386ee7	Grade 2
11	a33c38700f	Grade 3
12	a33c387079	Grade 3
13	6627e911d56	Grade 2
14	a3730e8420	Grade 4
15	a373a71a3d	Grade 3
16	a3743fb055	Grade 2
17	a3743fb121	Grade 3
18	a374d84676	Grade 3
19	a374d848cb	Grade 2
20	a374d8492a	Grade 2
21	a37570ddbe	Grade 3
22	a376097504	Grade 3
23	a3760975cd	Grade 3
24	a3aea94dbf	Grade 3
25	a376a209ee	Grade 2
26	a376a20b1e	Grade 5
27	a376a20b83	Grade 3
28	a3773aa0da	Grade 4
29	a3b1a441bd	Grade 5
30	a3773aa19c	Grade 1

4.2. Neural Network

A neural network algorithm is a computational gaining knowledge of gadget that uses a community of features to apprehend and translate records enters of 1 shape into a favored output, typically in some other form. Neural networks are simply considered one of many gear and methods utilized in machine mastering algorithms. Neural networks are being carried out to many actual-existence problems these days, along with speech and photo reputation, unsolicited mail e-mail filtering, finance, and scientific prognosis. The neural network mastering set of rules rather learns from processing many labeled examples which might be supplied at some point of education and the use of this solution key to study what traits of the input are hard to construct the correct output. Once an enough quantity of examples have been processed, the neural community can begin to system new, unseen inputs and efficaciously return accurate consequences.

This idea can nice be understood with an example. Believe the "easy" problem of looking to decide whether or not a photo includes a cat. Whilst that is rather clean for a human to parent out, it is plenty harder to educate a computer to discover a cat in a photo the usage of classical techniques. Considering the various possibilities of how a cat may also look in a photo, writing code to account for each situation is nearly not possible. However the usage of gadget gaining knowledge of, and more specifically neural networks, this system can use a generalized technique to information the content in an image. The use of several layers of functions to decompose the photograph into information factors and statistics that a laptop can use, the neural community can start to become aware of trends that exist across the various, many examples that it approaches and classify pictures by using their similarities. After processing many education examples of cat pictures, the algorithm has a model of what factors, and their respective relationships, in an photo are vital to recollect when figuring out whether or not a cat is present inside the photograph or not. [11]

The output of the neural network algorithm is shown as:

In [14]:

```
model = Sequential()
model.add(Dense(50, activation='tanh', kernel_initializer='random_uniform',
                bias_initializer='zeros', input_shape=(X_train_os.shape[1],)))
# model.add(Dropout(0.3))
model.add(Dense(40, activation='tanh'))
# model.add(Dropout(0.2))
model.add(Dense(30, activation='tanh'))
# model.add(Dropout(0.2))
model.add(Dense(30, activation='tanh'))
# model.add(Dropout(0.2))
model.add(Dense(30, activation='tanh'))
# model.add(Dropout(0.2))
model.add(Dense(30, activation='tanh'))
# model.add(Dropout(0.3))
model.add(Dense(5, activation='softmax'))
model.summary()
```

```
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorflow/python/framework/op_def_library.py:263: colocate_wit
h (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
```


Fig 5: Creating a Neural network model

```
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 50)                2500
_________________________________________________________________
dense_2 (Dense)              (None, 40)                2040
_________________________________________________________________
dense_3 (Dense)              (None, 30)                1230
_________________________________________________________________
dense_4 (Dense)              (None, 30)                930
_________________________________________________________________
dense_5 (Dense)              (None, 30)                930
_________________________________________________________________
dense_6 (Dense)              (None, 30)                930
_________________________________________________________________
dense_7 (Dense)              (None, 5)                 155
=================================================================
Total params: 8,715
Trainable params: 8,715
Non-trainable params: 0
_________________________________________________________________
```

Fig 6: Dense layers created in Neural network model

```
In [15]:
sgd = optimizers.SGD(lr=0.000001, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(optimizer = sgd, loss = "categorical_crossentropy", metrics = ["accuracy"])

In [16]:
model.fit(X_train_os, y_train_os_one_hot, epochs= 3, batch_size = 1000)

WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorfl
ow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Epoch 1/3
1021990/1021990 [==============================] - 9s 8us/step - loss: 1.6279 - acc: 0.2274
Epoch 2/3
1021990/1021990 [==============================] - 6s 6us/step - loss: 1.6110 - acc: 0.2349
Epoch 3/3
1021990/1021990 [==============================] - 6s 6us/step - loss: 1.6036 - acc: 0.2391

Out[16]:
<keras.callbacks.History at 0x7f761429d2b0>
```

Fig 7: Accuracy prediction of Neural network model

4.3. Xgboost

XGBoost is an ensemble studying approach. From time to time, it may no longer be enough to depend on the outcomes of just one machine learning model. Ensemble getting to know gives a scientific solution to combine the predictive strength of more than one beginner. The ensuing is a single version which gives the aggregated output from numerous models.

The fashions that shape the ensemble, additionally known as base newbies, could be both from the identical mastering set of rules or one of a kind gaining knowledge of algorithms. Bagging and boosting are widely used ensemble newbies. Even though those two techniques can be used with numerous statistical fashions, the maximum principal usage has been with decision bushes. [12]

Bagging

Even as decision timber is one of the maximum without difficulty interpretable models, they show off surprisingly variable behavior. Bear in mind a single education dataset that we randomly break up into two elements. Now, allows use each element to train a selection tree with a purpose to acquire two models.

When we match each these models, they could yield special consequences. Choice trees are said to be associated with excessive variance because of this conduct. Bagging or boosting aggregation facilitates to reduce the variance in any learner. Several decision bushes which are generated in parallel shape the base beginners of bagging method. Statistics sampled with substitute is fed to these rookies for education. The final prediction is the averaged output from all the freshmen.

Boosting

In boosting, the timber are built sequentially such that each subsequent tree ambitions to reduce the mistakes of the previous tree. Each tree learns from its predecessors and updates the residual errors. Subsequently, the tree that grows next within the sequence will analyze from an updated model of the residuals.

The base newcomers in boosting are weak newcomers in whom the prejudice is high, and the predictive power is just a tad higher than random guessing. Every of those weak beginners contribute some vital statistics for prediction, allowing the boosting approach to provide a sturdy learner by means of efficiently combining these vulnerable newbies. The very last sturdy learner brings down each the bias and the variance.

In assessment to bagging techniques like Random forest, wherein trees are grown to their maximum volume, boosting uses timber with fewer splits. Such small bushes, which aren't very deep, are quite interpretable. Parameters just like the variety of bushes or iterations, the price at which the gradient boosting learns, and the intensity of the tree, can be optimally selected via validation strategies like okay-fold cross validation. Having a massive wide variety of bushes may lead to overfitting. So, it's miles important to carefully select the stopping criteria for boosting.

The output of the xgboost algorithm is shown as:

Modelling

- Uncomment all the below cells and run, which reproduces the score on the leaderboard.

In [28]:
```
# xgb = xgboost.XGBClassifier(n_estimators=1000, learning_rate=0.1, subsample=0.7, colsample_bytree=1)
# model = xgb.fit(X_train_os, y_train_os_categories, verbose=1000)
```

In [ ]:
```
# model
```

In [41]:
```
# d_train = xgboost.DMatrix(X_train_os, label = np.array(y_train_os_categories.damage_grade - 1))
```

In [14]:
```
'''
params = {}
params['learning_rate'] = 0.1
params['booster'] = 'gbtree'
params['objective'] = 'multi:softmax'
params['metric'] = 'mlogloss'
params['max_depth'] = 20
params['num_class'] = 5
params['colsample_bytree'] = 1
params['subsample'] = 0.7
'''
```

Out[14]:
```
"\nparams = {}\nparams['learning_rate'] = 0.1\nparams['booster'] = 'gbtree'\nparams['objective'] = 'multi:softmax'\nparams['metric'] = 'mlogloss'\nparam
s['max_depth'] = 20\nparams['num_class'] = 5\nparams['colsample_bytree'] = 1\nparams['subsample'] = 0.7\n"
```

In [ ]:
```
# xgbClassifier = xgboost.train(params, d_train, 5000)
```

Fig 8: Code of Xgboost algorithm

- Save the model file.

In [ ]:
```
import pickle
pkl_filename = "xgboost_latest.pkl"
with open(pkl_filename, 'wb') as file:
    pickle.dump(model, file)
```

In [ ]:
```
model
```

In [32]:
```
y_train_pred = model.predict(X_train_os)
```

In [36]:
```
y_train_pred.shape, y_train_os.shape
```

Out[36]:
```
((1021990,), (1021990, 1))
```

In [41]:
```
y_pred_np = np.array(y_train_os_categories.damage_grade)
```

In [42]:
```
Accuracy = accuracy_score(y_pred_np, y_train_pred) * 100
F1_score = f1_score(y_pred_np, y_train_pred, average='weighted')
```

In [43]:
```
Accuracy, F1_score
```

Out[43]:
```
(97.52580749322401, 0.9752548313724714)
```

Fig 9: Accuracy prediction of Xgboost algorithm

- Feature importance of all top features.

In [44]:
```
feature_importance_df = feature_importance(model, X_train_os)
feature_importance_df.iloc[:35].plot('bar')
```

Out[44]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f21b8d48940>
```

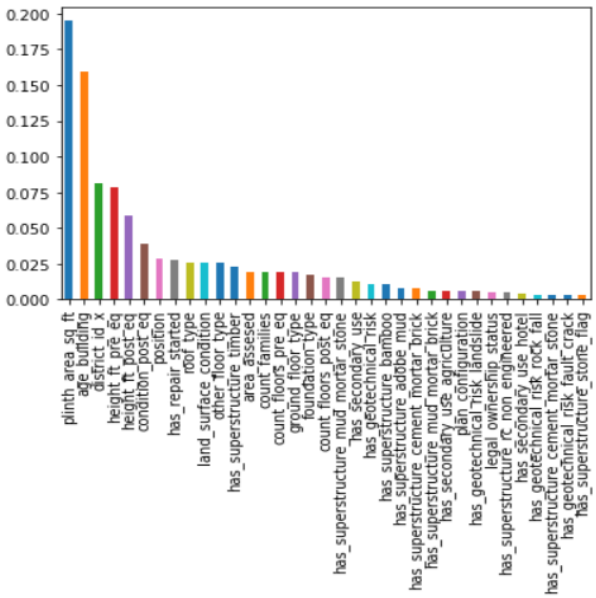

Fig 10: Feature importance of attributes using Xgboost algorithm

5. CONCLUSION

Accordingly we can finish that integration of seismic activity with device learning generation yields efficient and tremendous end result and can be used to expect earthquakes extensively; given the beyond history of the same is nicely maintained. Our try can be termed a hit. The collaboration of the two can in addition be superior to guard earthquakes more acutely. Large datasets prove to be very big. Prediction models may be deployed in a place-centric manner, therefore increasing the chances of correct prediction exponentially however on the fee of analyzing algorithms used to build Stacking version, as it will carry out well only if the algorithms chosen to build metaregressor are correct themselves. The use of the technique may be extended in predicting diverse herbal screw ups as well. [13]

References

[1] The Expert Team. “What is Machine Learning? A definition”, Expertsystem, 7 March 2017.

[2] Nick Minaie. “A Beginner’s Guide to Selecting Machine Learning Predictive Models in Python”, Towardsdatascience, Medium, 16 July 2019.

[3] Cover, T.; Hart, P. Nearest neighbor pattern classification. IEEE Trans. Inf. Theory 1967, 13, 21–27.

[4] Rokach, L.; Maimon, O. Decision trees. In Data Mining and Knowledge Discovery Handbook; Springer: Berlin, Germany, 2005; pp. 165–192.

[5] Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32.

[6] Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y. Xgboost: Extreme gradient boosting. In Microsoft. R Package Version 0.4-2; R Package Vignette: Madison, WI, USA, 2015; pp. 1–4.

[7] Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. Adv. Neural Inf. Proces. Syst. 2017, 30, 3146–3154.

[8] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 2011, 12, 2825–2830.

[9] Kumar, Vivek. “Vivek Kumar.” Pluralsight, 13 May 2019, www.pluralsight.com/guides/building-classification-models-scikitlearn.

[10] Flach, Peter. Machine Learning: the Art and Science of Algorithms That Make Sense of Data. Cambridge University Press, 2017, pp. 331-333.

[11] https://deepai.org/machine-learning-glossary-and-terms/neural-network.

[12]https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/.

[13] Earthquake Prediction using Machine Learning Algorithm By Pratiksha Bangar, Deeksha Gupta, Sonali Gaikwad, Bhagyashree Marekar, Jyoti Patil.