Research Article

A Comparison Experiment On Software Quality Testing With Machine Learning Using Different Algorithms

Munidhanalakshmi Kumbakonam[1], Mahammad Shafi RajaSaheb[2]

[1]Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore, Tamilnadu, India.
kmd.dhana@gmail.com
[2]Professor, Department of MCA, Sree Vidyanikethan Engineering College, Tirupati, Andrapradesh, India.
rmdshafi@gmail.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

ABSTRACT
The software quality is a particular property that indicates the regular software. The main success factor and departure of software-related organisations in a software project is efficiency. Machine-based standard deviation and mean use of CMMI as a software quality assurance model – which in computer software are structured in terms of objectives, obligations, capacities, tasks, measurement and above all validations via the engineer-based application of ISO 9000/9001-3. Different algorithms such as Random Forest, Decision Tree, Gradient Boosting, Bagging Classifier, Logistic Regression, Bernoulli-NB and CNN are used in this paper to predict software qualities. It contributes to solving and handling a program's time complexity and enhances software efficiency, reliability and security issues.

KEYWORDS: SQA, CMMI, CNN, TIME COMPLEXITY, SOFTWARE PERFORMANCE.

1. INTRODUCTION

Software testing is a method that tests for optimal performance and quantity of software. In the environment it has been designed, the basic test of the programme takes place. It is run and the right and successful outputs are checked. Software research is also conducted in international environments to investigate scalability options[8, 12]. In different strategic environments, each software is reviewed.

Load tests consist of performance tests which are based on the identification or validation of the product under examination while subject to workload models and expected load volumes during manufacturing. Load tests are a mandatory examination before any test is performed. In general, a test case is a data used as an input for testing of program. This includes a unique identifier, program specification criteria references, a sequence of follow up steps, events, circumstances, input, output, expected outcome and actual outcome.

The tests carried out should yield correct results, but all shortcomings cannot be detected at any time under the assumptions of specific functions. Rather, it compares the state and actions, by means of which users may understand the problem, to the product principles/mechanisms.

Leistungstests are used to assess end-to-end output accurately. It involves the calculation of the response time of the end user, consistent load repeating, managed load monitoring of system components, solid analytical and reporting engines.

A. CMMI levels:

A process and comportability model that enables organisations to streamline process creation and to foster effective and efficient conduct that reduce software, product and service development risks is called CMMI (Capability-Maturity-Model-Integration).

The CMMI model divides corporate maturity into five stages. The aim is to increase the company to level 5, "optimising" maturity, for enterprises which use CMMI. Once companies hit this stage, the CMMI is not completed. They are focused instead on repairs and routine upgrades.

B. CMMI's five Maturity Levels are:

a. **Initial:** Volatile and reactive processes are considered. "The work is done at this point, but it is always overdue and overdue." This is the worst process in which an organisation can be found—a condition that raises risk and inefficiency.
b. **Managed:** The project management standard has been reached. Projects at this stage have been "designed, implemented, assessed and monitored," but a lot of problems have been resolved.
c. **Defined:** organisations are constructive rather than reactive at this point. There are a number of organisational guidelines for "guiding initiatives, services and portfolios." Enterprises know their weaknesses, how to deal with them and what the goal is**.**
d. **Handled quantitatively:** More calculated and supervised, this phase. The company works on objective data to identify predictable and stakeholder-compatible processes. The organisation faces risks with an insight into process defects by more details.
e. **Optimisation:** The procedures of an enterprise are robust and scalable here. An organisation will continuously develop and adapt to changes or other opportunities at this final level. The organisation, in a predictable setting, is stable, providing more "agility and creativity."

Once organisation levels 4 and 5 have been achieved, it is considered to be highly mature, where it "continuously evolves and adapts to meet stakeholder and consumer needs and grows." That is the objective of the CMMI: to build reliable environments that provide constructive, effective and productive goods, services and divisions..

2. LITERATURE REVIEW

A. Ozgür TANRIÖVER et. Ayberk CERAN et al. An experimental study with machine-learning methods for predicting software quality. The feature selection method and correlation matrix are used in this method author to achieve higher precision and increase estimation precision by the use of the related characteristics of a large dataset. Recent approaches have been validated that have proven effective for other prediction tasks. The data used to predict software quality and to reveal a relationship between the attributes of quality and growth are machine learning algorithms such as xgboost, random forest and decision tree.

Akshat Sharma et. al Software Testing Use Genetic Algorithms, and Rishon Patani and Ashish Aggarwal. In the present text, a series of methods is presented, which utilises a genetic algorithm for the automated generation of test data during software testing, and the test methods use different genetic algorithms to analyse test populations using structurally-orienteed test data.

Gail Kaiser and Marta Arias et al, Christian Murphy et. Machine Learning Applications Quality Assurance System. The author describes in this paper a system for testing and debugging of regulated ML applications implementing ranking algorithms. ML implements the generator for test data set to obtain performance models and classification as (Marti Rank).

Sangeeta Sabharwal, Chayanika Sharma, Sibal and others. A study of genetic algorithm-based software research techniques. In this approach, the end user is guaranteed high-quality software to conduct software tests using GA effectively to increase testing time and costs, and GA used in various types of tests with fluid as well as in the neural systems.

Ibrahim Dyana Rashid, Ghnemat Rawan, Hudaib Amjad et al. Prediction for defects of software with Random Forest Algorithms and Feature Selection. This paper describes two algorithms that are Bat Baes study algorithms for the selection of features and the random forest to assess the efficiency of the quality assurance of applications.

Shaik Hazar Babavali, K.Rahul, N.V. Vishnu Teja, E. Sri Devi et al. Software Defect Detection Using Decision Tree Learning Algorithms. This Paper Uses Different types of Decision Tree Algorithms for Accurate Defect Rate Analysis. It helps to use identify the different types of defects in software and improve the advance method using decision tree. Machine learning algorithm uses defects and non-defect module to set the average data set.

3. PROPOSED CAPABILITY-MATURITY-MODEL-INTEGRATION LEVELS

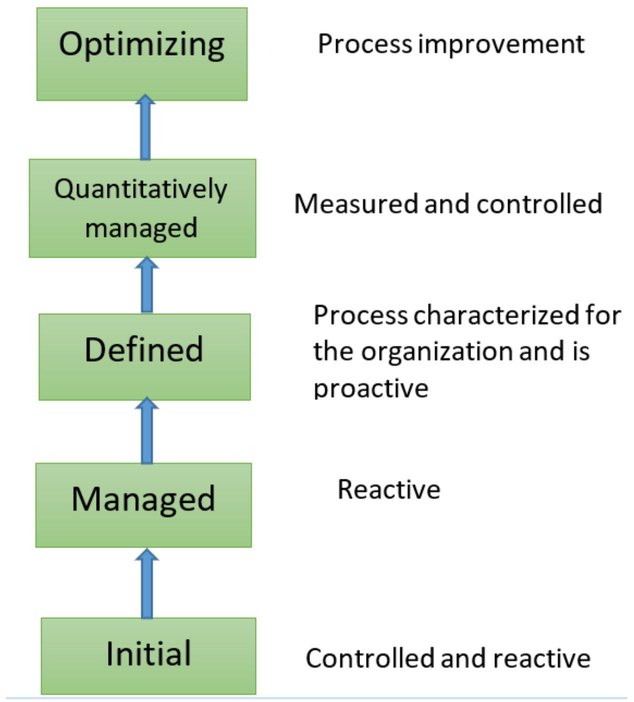


Fig1. CMMI LEVELS

In the proposed method we use to develop the CMMI techniques. Prepare SQA plan for the project to maintain organization by training, testing, review, and risk management to check the percentage of time complexity controlled by verification and validation. In this project we are proposed various machine learning algorithms such as Random Forest, Decision Tree, Gradient Boosting, Bagging Classifier, Logistic Regression, Bernoulli NB and CNN to predict software quality. To implement this project, we proposed two datasets i.e.; ISBSG and EBSPM. This dataset saved inside Dataset folder. Software testing is one of the major and primary techniques for achieving high quality software. The prime goal is to ensure the quality and time management.

4. METHODOLOGY

The experiment is conduct during python tool. The proposed model used EBSPM and ISBSG Datasets. The data set contain 30 important feature contain 36928 records in which few attributes contain NULL values and string non numeric values and we need to replace all missing and non-numeric values with their count. Load the input and update / selection file to test and train the data set. And to build the model by CNN to evaluate the test results or values to calculate the accuracy, Precision, Recall for each comparison algorithm.

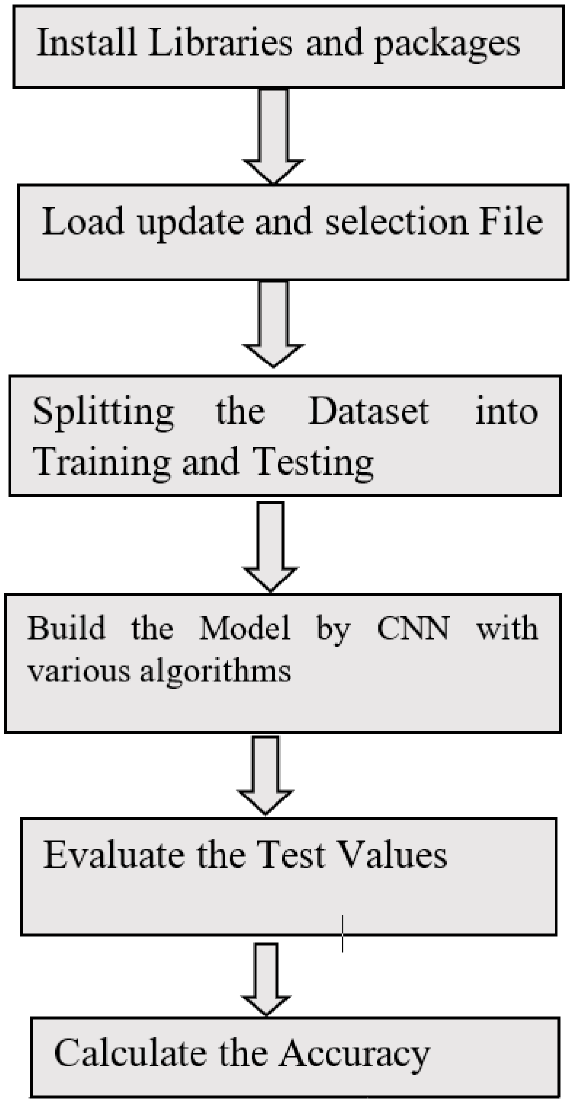

Install Libraries and packages
Load update and selection File
Splitting the Dataset into Training and Testing
Build the Model by CNN with various algorithms
Evaluate the Test Values
Calculate the Accuracy

Fig2. Implementation process of proposed Model.

5. EXISTING METHOD

To detect the existence of faults that cause software failure, software tests are performed. It absorbs nearly 50 percent of software development energy, as no validation, regression, models and other related models demonstrate proper performance. It is usually not easy to find out whether or not the output of the programme is "right" to the input.

6. PROPOSED TESTING ALGORITHMS

A. Gradient Boosting

In Gradient Boosting, each of the predictor tries to improve on its predecessor by reducing errors. However, captivating thought behind Inclination Boosting is that as opposed to fitting an indicator on the information at every emphasis, it really fits another indicator to the lingering blunders made by the past indicator with respect to dataset.

In order to make initial predictions on data to test values using comparison of algorithms. We calculate GB through the number of true values divided by number of false values to get the equivalent number which we got the accuracy value is 96%.

B. Bagging classifier

Bagging utilizes the straightforward methodology that appears in the factual investigations over and over-improve gauge of one by consolidating appraisals of numerous, it tests dataset that predicts incorrect data and create D>1.0 to be estimated the values to accuracy, recall and precision and repeated n times to get a proper test value.

C. Random forest

Random forest adds additional randomness to the model, Rf algorithm is applicable for both classification and regression. It handles the missing qualities; it can't over fit the mode when quantities of trees are expanded.
Random selection is calculated through n<<N to get the better characteristics to split the method, it evaluates the test data and estimate the missing values and to get the progressed data with compared datasets and provides large accuracy.

D. Decision tree

DT provides a specific test data to different attributes as well as evaluating test datasets, it creates a test data using entropy for the better results and it will traverse through the starting root node.
The resultant data will get the maximum accuracy to get the large amount of test data.

E. Convolutional Neural Networks

As we know, it is a learning engine algorithm to understand the characteristics of the image in advance and remember the function to devise whether the computer is supplied with the name of the new image. (CNNs) are the most common image classification neural network model. The brilliant idea behind CNNs is that a picture is good enough for the local community. The functional advantage is that less parameters dramatically increase learning time and reduce the amount of data needed to train the model. The CNN has only enough masses to examine a small piece of the picture instead of a completely linked network of weights from each pixel.

A concentration is a weighted number of the image's pixels, as the window slides over the entire image. This method results in a weight matrix picture creating another picture (of the same size, depending on the convention). Participation is the way to implement a convolution. There are many convolution layers in a typical CNN. Therefore, the weight matrix is 5 to 5 tensors per n, and n is the number of turns. There are multiple alternating turnouts in each convolution.

7. EBSPM AND ISBSG DATASETS

Technology Portfolio Management (EBSPM) is an evidence-based, realistic research repository that supports software companies in their actively improving software delivery portfolio. In order to promote software development creativity of an organisation, the EBSPM aims at evaluating, assessing and benchmarking interconnected software projects with respect to measurements, prices, length and the number of defects.
ISBSG has the aim of promoting the use of information from the IT sector to develop software and goods.

8.DATA PROCESSING

In data sets, we neglect values. For training accuracy it is necessary to delete some rows with missing and unknown values. The software output prediction using defect amounts in ISBGS datasets has two directly comparable studies. With several null values, both data sets were credited. Therefore, the attributes with fewer empty lines were chosen as predictor characteristics. Dataset has been reduced to 11few attributes and 1 target class following this procedure. The data set of EBSPM was reduced to the minimum attributes and 1 aim class.

9. RESULTS:

Click on the 'run.bat' file to double the project underneath the screen

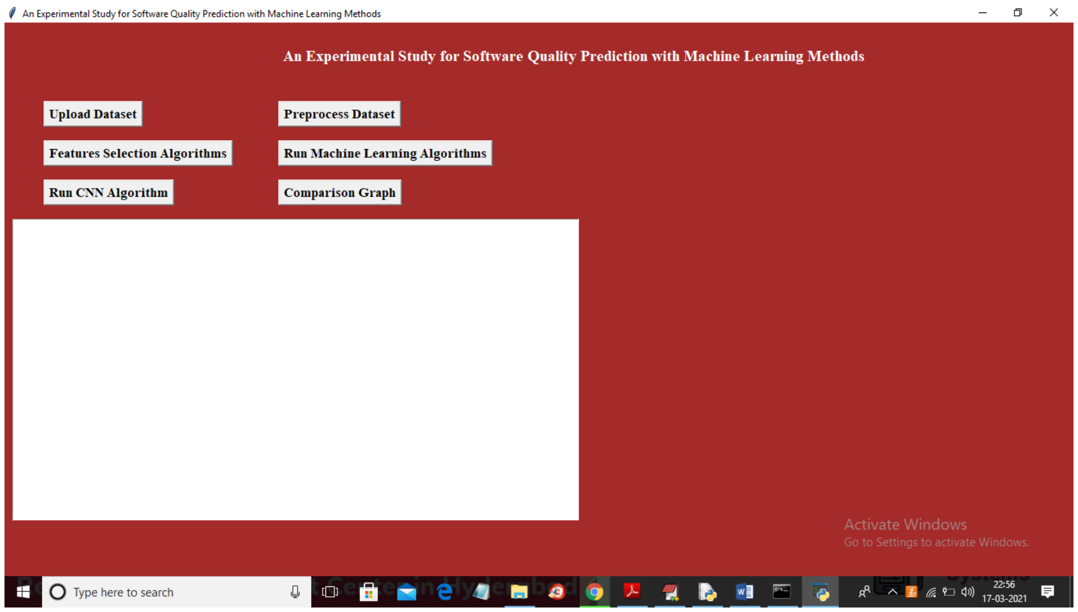

An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
22:56
17-03-2021

Fig3. Click on the 'Data Set Upload' button in the above screen.

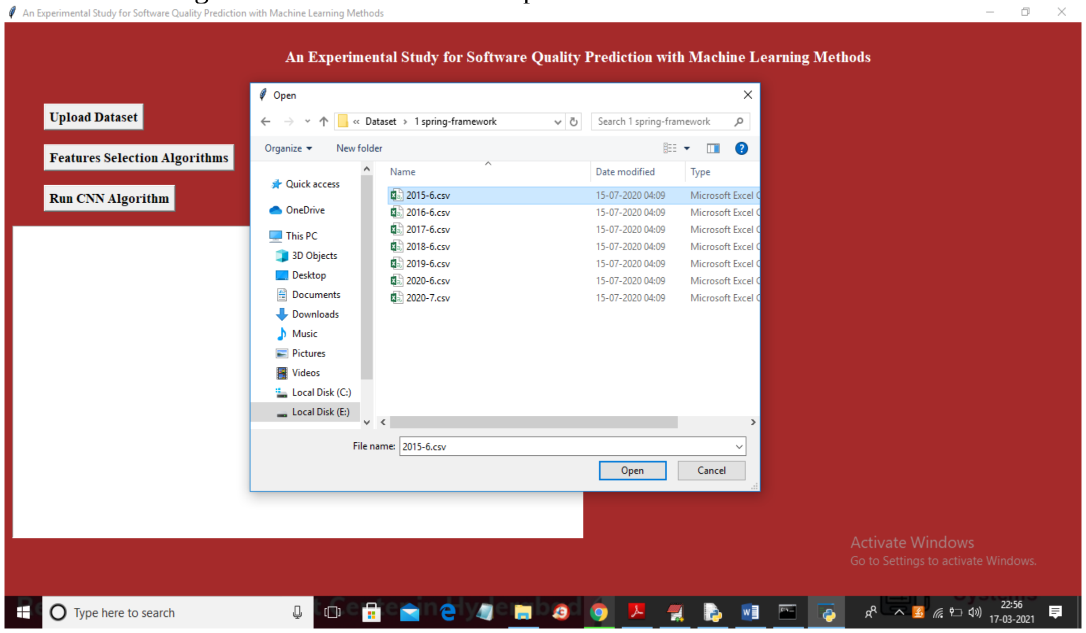

An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Features Selection Algorithms
Run CNN Algorithm
Open
Dataset › 1 spring-framework
Search 1 spring-framework
Organize
New folder
Quick access
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Name
Date modified
Type
2015-6.csv 15-07-2020 04:09 Microsoft Excel
2016-6.csv 15-07-2020 04:09 Microsoft Excel
2017-6.csv 15-07-2020 04:09 Microsoft Excel
2018-6.csv 15-07-2020 04:09 Microsoft Excel
2019-6.csv 15-07-2020 04:09 Microsoft Excel
2020-6.csv 15-07-2020 04:09 Microsoft Excel
2020-7.csv 15-07-2020 04:09 Microsoft Excel
File name: 2015-6.csv
Open
Cancel
Activate Windows
Go to Settings to activate Windows.
Type here to search
22:56
17-03-2021

Fig4. In the screen above, pick and upload the data set file '2015-6.csv,' and then click on the 'Open' button to load the dataset.

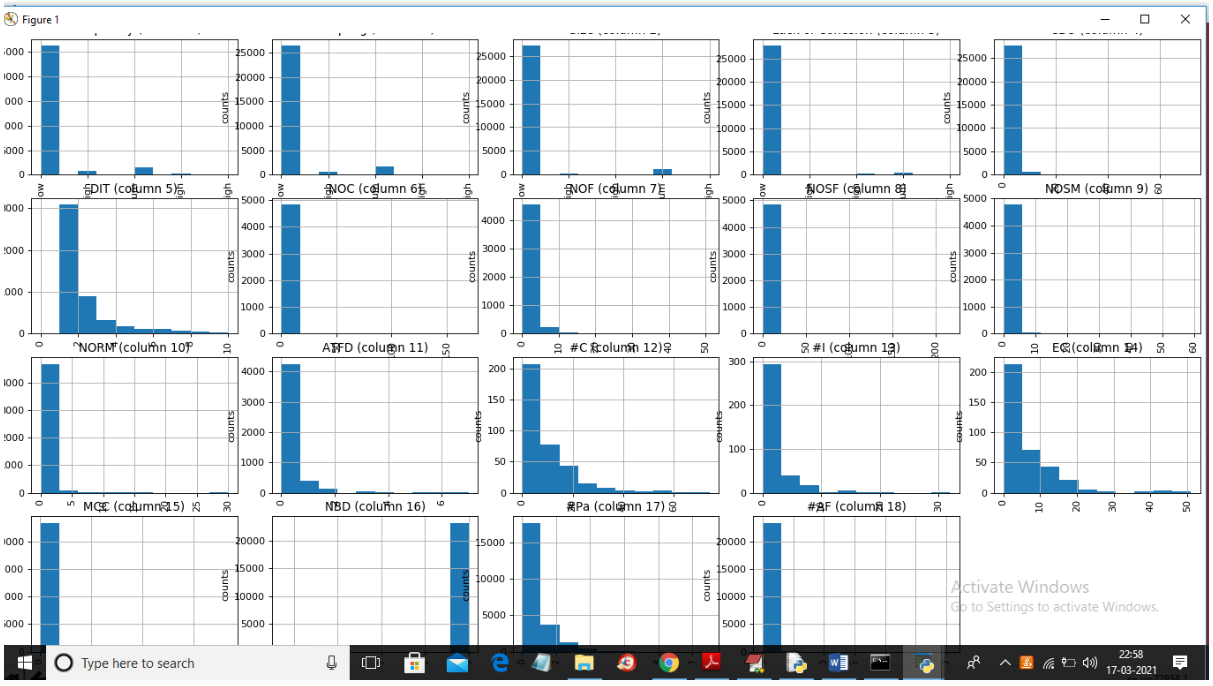
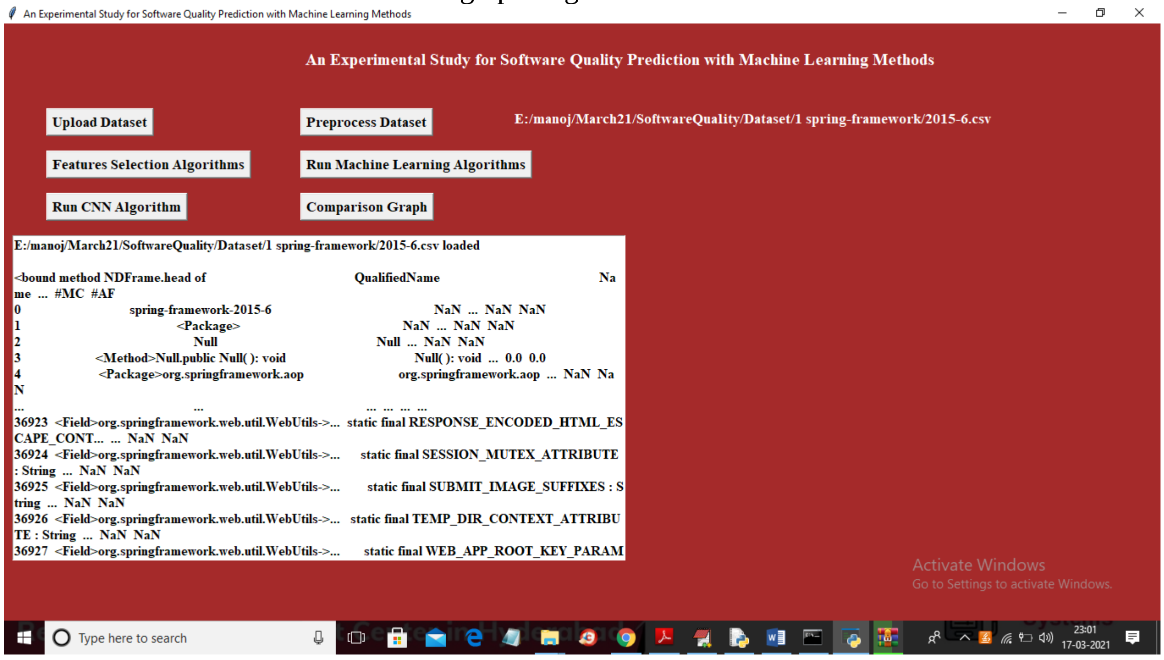


Fig5. In above graph we can see each graph represents one column from dataset and from that columns its counting each distinct value from and plot in that graph for example in second graph NOC columns 3 different values and its plotting 3 different bars with count and no close above graph to get below screen


An Experimental Study for Software Quality Prediction with Machine Learning Methods
An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph

E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv

```
E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv loaded

<bound method NDFrame.head of                               QualifiedName                Na
me  ... #MC  #AF
0                spring-framework-2015-6                     NaN  ...  NaN  NaN
1                           <Package>                    NaN  ...  NaN  NaN
2                              Null                   Null  ...  NaN  NaN
3             <Method>Null.public Null( ): void              Null( ): void  ...  0.0  0.0
4              <Package>org.springframework.aop            org.springframework.aop  ...  NaN  Na
N
...                                 ...                  ...  ...  ...  ...
36923  <Field>org.springframework.web.util.WebUtils->...  static final RESPONSE_ENCODED_HTML_ES
CAPE_CONT...  ...  NaN  NaN
36924  <Field>org.springframework.web.util.WebUtils->...    static final SESSION_MUTEX_ATTRIBUTE
: String  ...  NaN  NaN
36925  <Field>org.springframework.web.util.WebUtils->...     static final SUBMIT_IMAGE_SUFFIXES : S
tring  ...  NaN  NaN
36926  <Field>org.springframework.web.util.WebUtils->...   static final TEMP_DIR_CONTEXT_ATTRIBU
TE : String  ...  NaN  NaN
36927  <Field>org.springframework.web.util.WebUtils->...     static final WEB_APP_ROOT_KEY_PARAM
```

Fig6. In above screen displaying values from dataset and we can see dataset contains NAN (missing values) and string non numeric values and we need to replace all missing and non-numeric values with their count so click on ‘Preprocess Dataset’ button

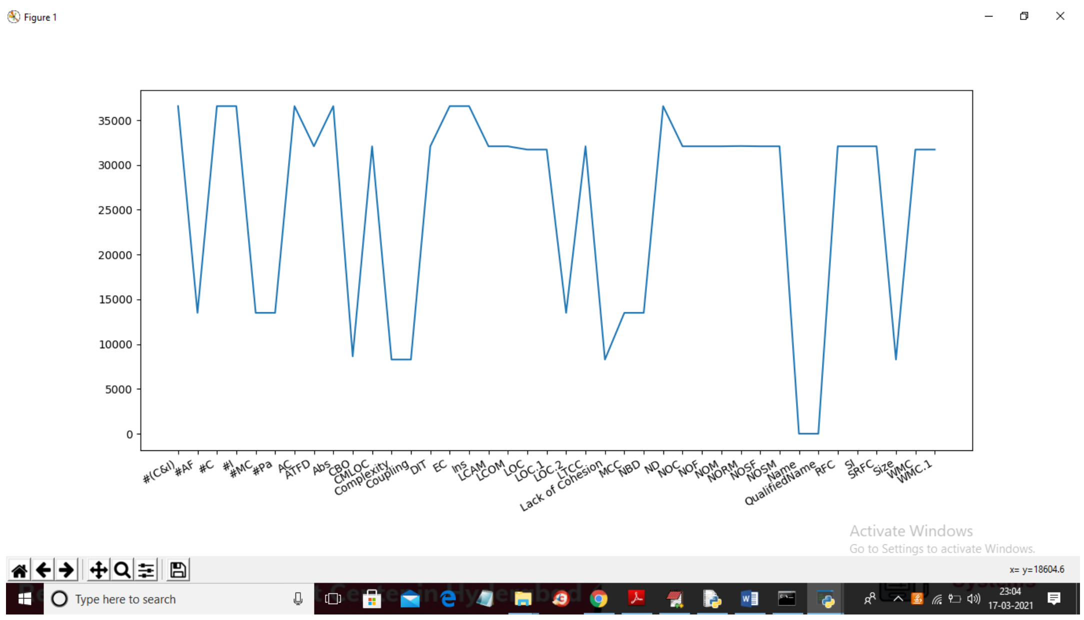
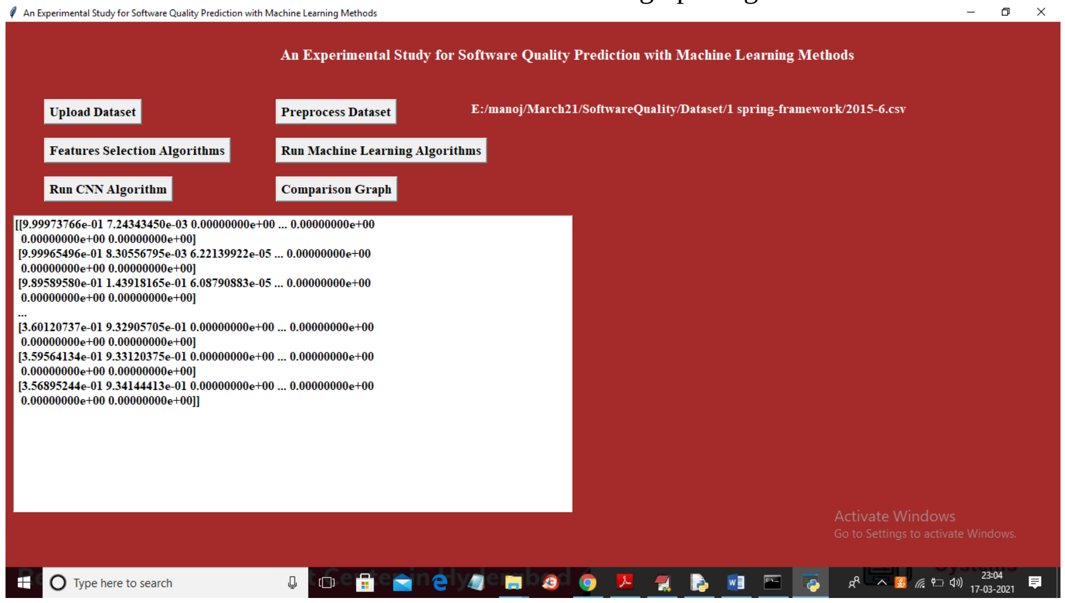


Fig7. In above graph x-axis represents column names and y-axis represents total missing values counts in that column and now close above graph to get below screen


An Experimental Study for Software Quality Prediction with Machine Learning Methods
An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Type here to search
23:04
17-03-2021

```
[[9.99973766e-01 7.24343450e-03 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [9.99965496e-01 8.30556795e-03 6.22139922e-05 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [9.89589580e-01 1.43918165e-01 6.08790883e-05 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 ...
 [3.60120737e-01 9.32905705e-01 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [3.59564134e-01 9.33120375e-01 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [3.56895244e-01 9.34144413e-01 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]]
```

Fig8. In above screen all missing a string values are replaced with numeric values and now click on 'Features Selection Algorithms' button to select important features from dataset and then split dataset into train and test part

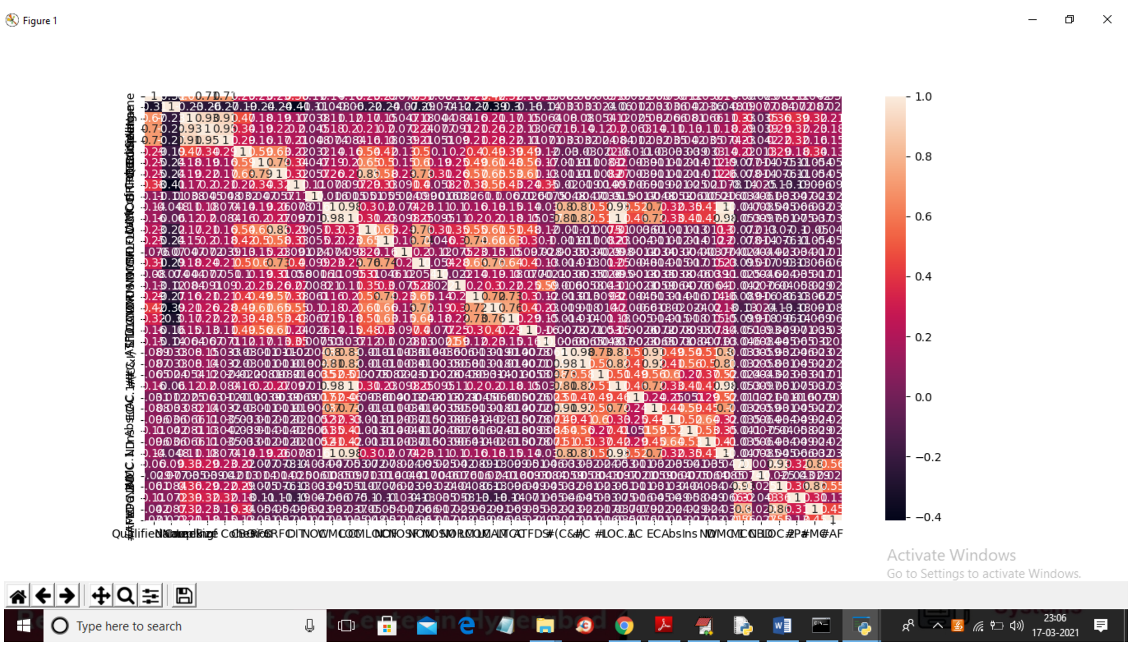
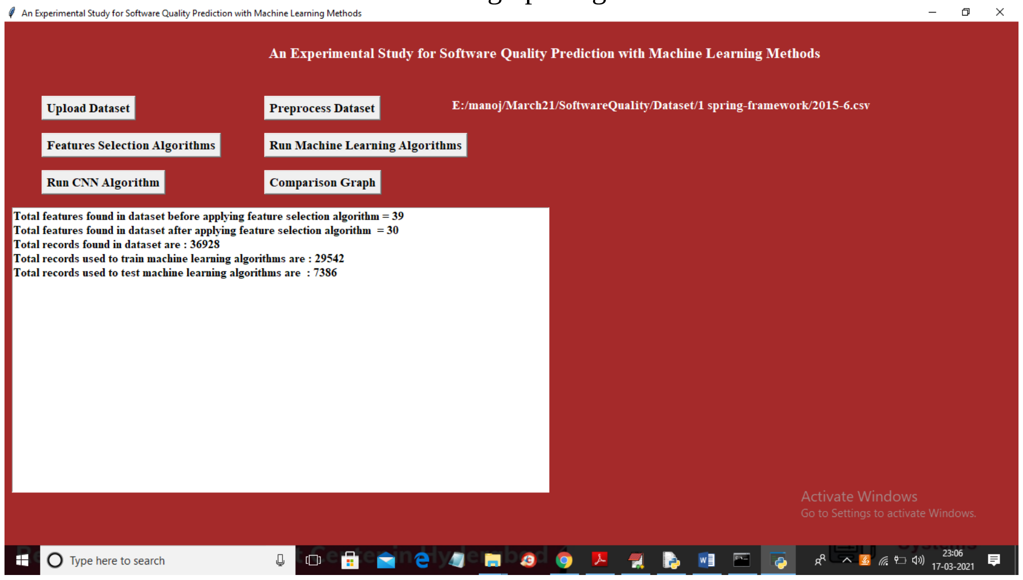


Fig9. In above graph the box which contains value >0.5 will be consider as important attributes and now close above graph to get below screen

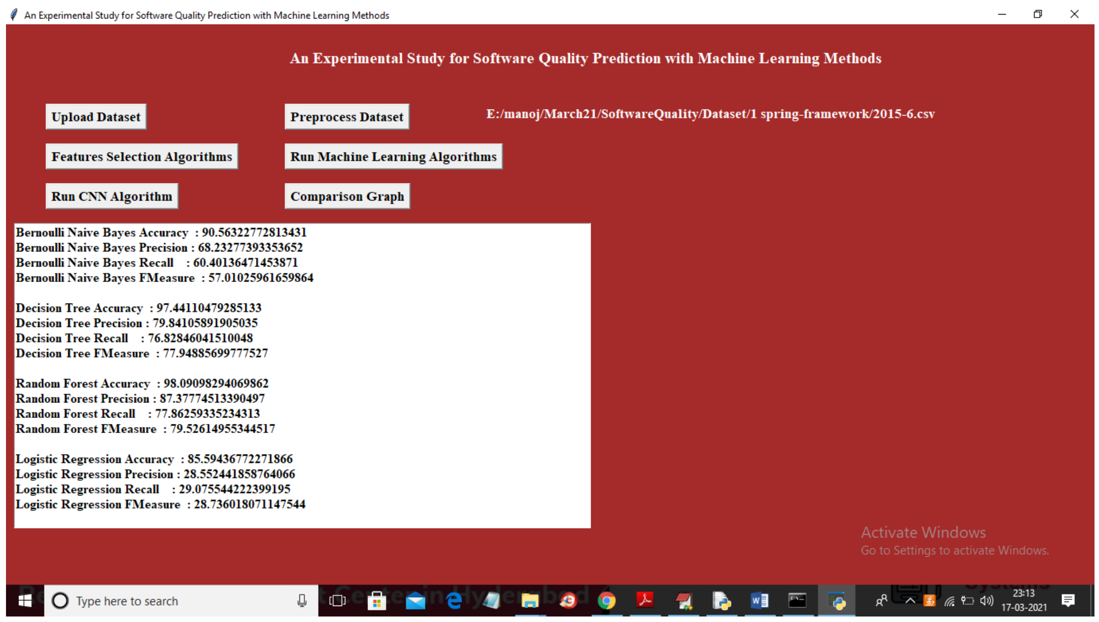

An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
23:06
17-03-2021

An Experimental Study for Software Quality Prediction with Machine Learning Methods

E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv

```
Total features found in dataset before applying feature selection algorithm = 39
Total features found in dataset after applying feature selection algorithm  = 30
Total records found in dataset are : 36928
Total records used to train machine learning algorithms are : 29542
Total records used to test machine learning algorithms are  : 7386
```

Fig10. In above screen before applying feature selection algorithm dataset contains 39 features/columns and after applying PCA feature selection we got 30 important features and dataset contains 36928 records and application using 7386 records for testing and 29542 records for training and now both train and test dataset is ready and now click on ‘Run Machine Learning Algorithms’ button to run all machine learning algorithms


An Experimental Study for Software Quality Prediction with Machine Learning Methods
An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Type here to search
23:13
17-03-2021

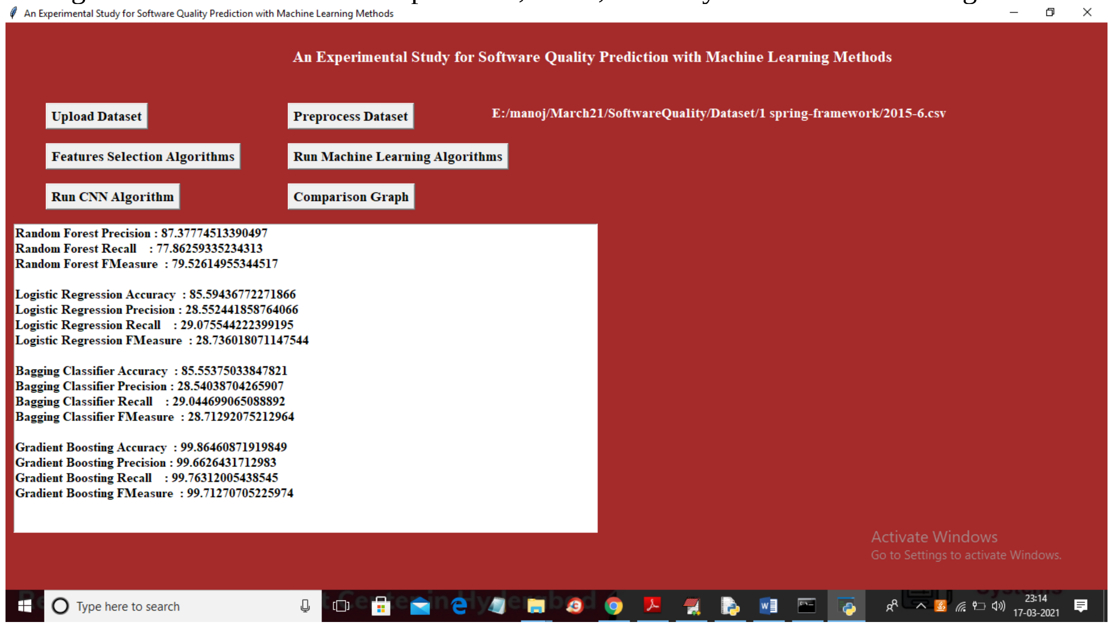
```
Bernoulli Naive Bayes Accuracy  : 90.56322772813431
Bernoulli Naive Bayes Precision : 68.23277393353652
Bernoulli Naive Bayes Recall    : 60.40136471453871
Bernoulli Naive Bayes FMeasure  : 57.01025961659864

Decision Tree Accuracy  : 97.44110479285133
Decision Tree Precision : 79.84105891905035
Decision Tree Recall    : 76.82846041510048
Decision Tree FMeasure  : 77.94885699777527

Random Forest Accuracy  : 98.09098294069862
Random Forest Precision : 87.37774513390497
Random Forest Recall    : 77.86259335234313
Random Forest FMeasure  : 79.52614955344517

Logistic Regression Accuracy  : 85.59436772271866
Logistic Regression Precision : 28.552441858764066
Logistic Regression Recall    : 29.075544222399195
Logistic Regression FMeasure  : 28.736018071147544
```

Fig11. In above screen we can precision, recall, accuracy and f-score for all algorithms


An Experimental Study for Software Quality Prediction with Machine Learning Methods
An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Type here to search
23:14
17-03-2021

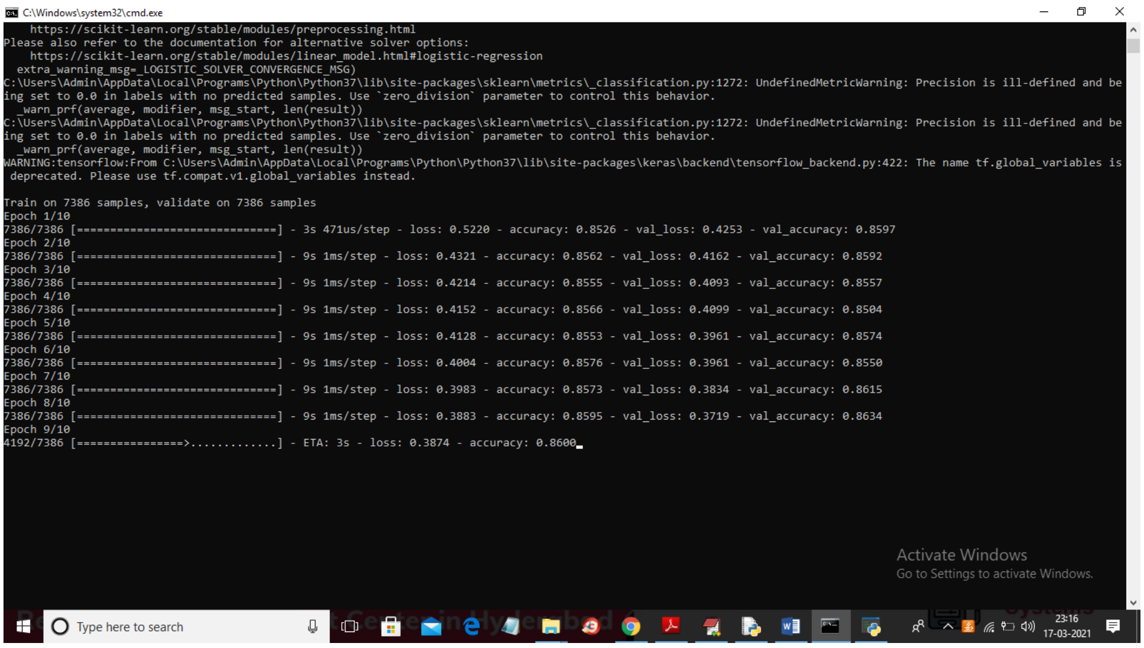
```
Random Forest Precision : 87.37774513390497
Random Forest Recall    : 77.86259335234313
Random Forest FMeasure  : 79.52614955344517

Logistic Regression Accuracy  : 85.59436772271866
Logistic Regression Precision : 28.552441858764066
Logistic Regression Recall    : 29.075544222399195
Logistic Regression FMeasure  : 28.736018071147544

Bagging Classifier Accuracy  : 85.55375033847821
Bagging Classifier Precision : 28.54038704265907
Bagging Classifier Recall    : 29.044699065088892
Bagging Classifier FMeasure  : 28.71292075212964

Gradient Boosting Accuracy  : 99.86460871919849
Gradient Boosting Precision : 99.6626431712983
Gradient Boosting Recall    : 99.76312005438545
Gradient Boosting FMeasure  : 99.71270705225974
```

Fig12. Now click on ‘Run CNN Algorithm’ button to run CNN algorithm and to get below screen

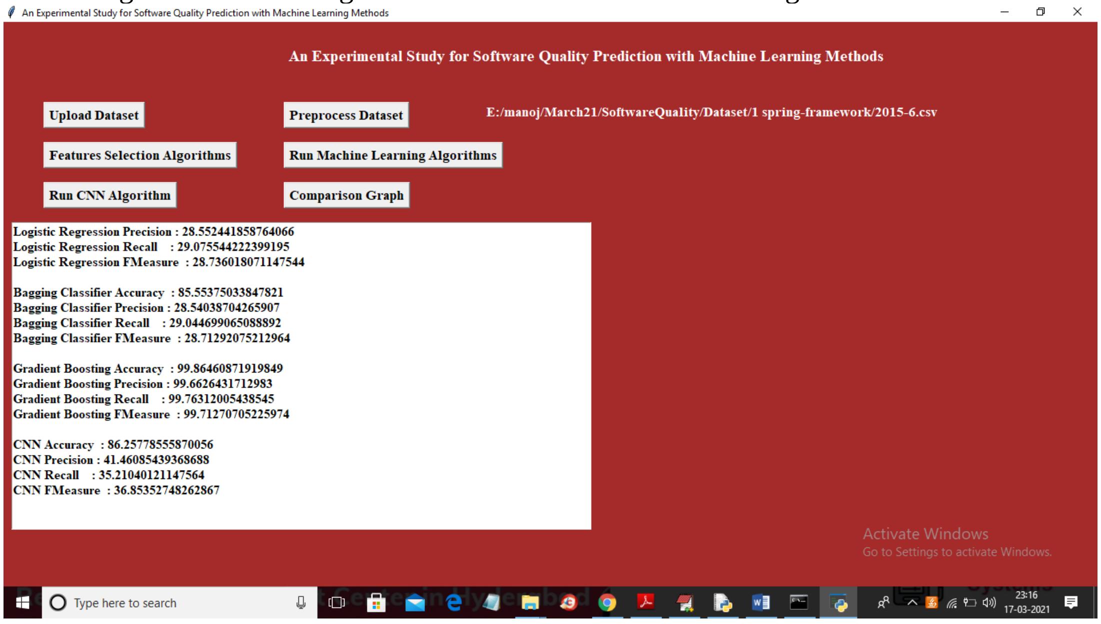


```
   https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
   https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\metrics\_classification.py:1272: UndefinedMetricWarning: Precision is ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\metrics\_classification.py:1272: UndefinedMetricWarning: Precision is ill-defined and be
ing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow_backend.py:422: The name tf.global_variables is
 deprecated. Please use tf.compat.v1.global_variables instead.

Train on 7386 samples, validate on 7386 samples
Epoch 1/10
7386/7386 [==============================] - 3s 471us/step - loss: 0.5220 - accuracy: 0.8526 - val_loss: 0.4253 - val_accuracy: 0.8597
Epoch 2/10
7386/7386 [==============================] - 9s 1ms/step - loss: 0.4321 - accuracy: 0.8562 - val_loss: 0.4162 - val_accuracy: 0.8592
Epoch 3/10
7386/7386 [==============================] - 9s 1ms/step - loss: 0.4214 - accuracy: 0.8555 - val_loss: 0.4093 - val_accuracy: 0.8557
Epoch 4/10
7386/7386 [==============================] - 9s 1ms/step - loss: 0.4152 - accuracy: 0.8566 - val_loss: 0.4099 - val_accuracy: 0.8504
Epoch 5/10
7386/7386 [==============================] - 9s 1ms/step - loss: 0.4128 - accuracy: 0.8553 - val_loss: 0.3961 - val_accuracy: 0.8574
Epoch 6/10
7386/7386 [==============================] - 9s 1ms/step - loss: 0.4004 - accuracy: 0.8576 - val_loss: 0.3961 - val_accuracy: 0.8550
Epoch 7/10
7386/7386 [==============================] - 9s 1ms/step - loss: 0.3983 - accuracy: 0.8573 - val_loss: 0.3834 - val_accuracy: 0.8615
Epoch 8/10
7386/7386 [==============================] - 9s 1ms/step - loss: 0.3883 - accuracy: 0.8595 - val_loss: 0.3719 - val_accuracy: 0.8634
Epoch 9/10
4192/7386 [================>.............] - ETA: 3s - loss: 0.3874 - accuracy: 0.8600
```

C:\Windows\system32\cmd.exe
Activate Windows
Go to Settings to activate Windows.
Type here to search
23:16
17-03-2021

Fig13. In above screen to train CNN we took 10 iterations or epoch and at each epoch accuracy get better and loss get reduce and after 10 iterations will get below screen



An Experimental Study for Software Quality Prediction with Machine Learning Methods
An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
23:16
17-03-2021

```
Logistic Regression Precision : 28.552441858764066
Logistic Regression Recall    : 29.075544222399195
Logistic Regression FMeasure  : 28.736018071147544

Bagging Classifier Accuracy  : 85.55375033847821
Bagging Classifier Precision : 28.54038704265907
Bagging Classifier Recall    : 29.044699065088892
Bagging Classifier FMeasure  : 28.71292075212964

Gradient Boosting Accuracy  : 99.86460871919849
Gradient Boosting Precision : 99.6626431712983
Gradient Boosting Recall    : 99.76312005438545
Gradient Boosting FMeasure  : 99.71270705225974

CNN Accuracy  : 86.25778555870056
CNN Precision : 41.46085439368688
CNN Recall    : 35.21040121147564
CNN FMeasure  : 36.85352748262867
```

Fig14. In above screen we got output values for CNN also and now click on 'Comparison Graph' button to get below screen.

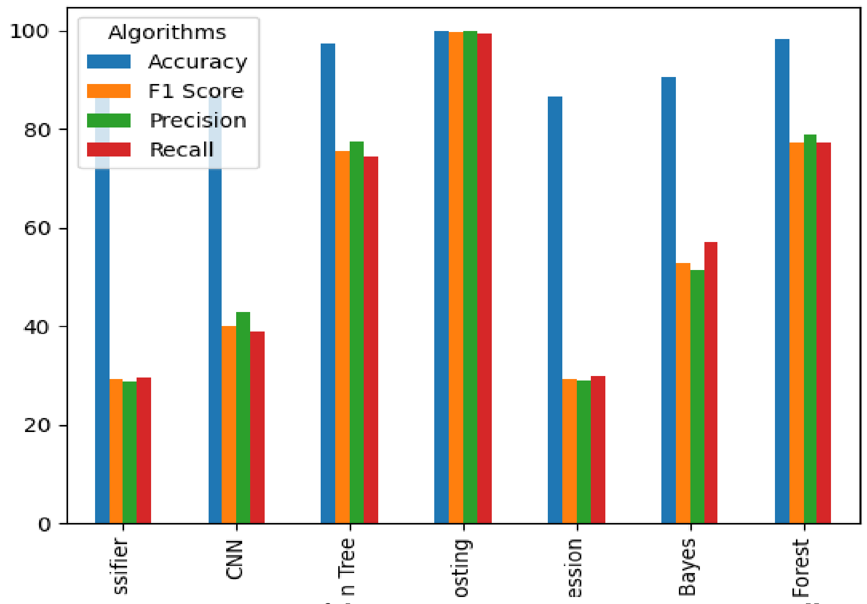


Fig15. Comparison of the Accuracy, F1 score, Precision, Recall.

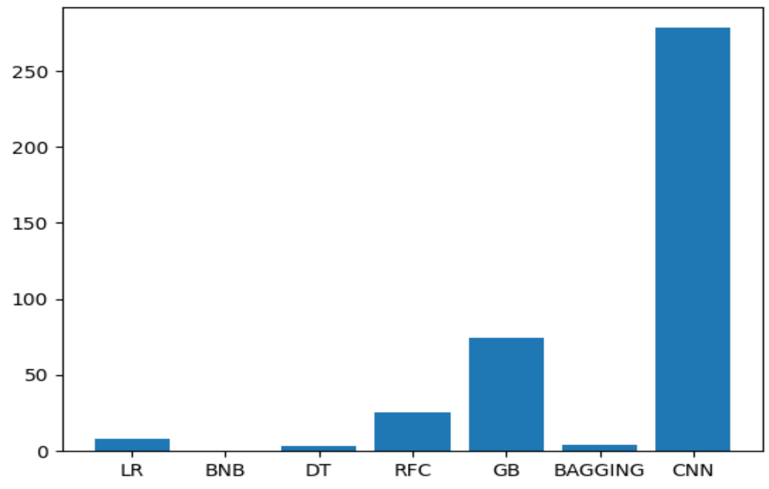


Fig16. Time comparison of the algorithms.

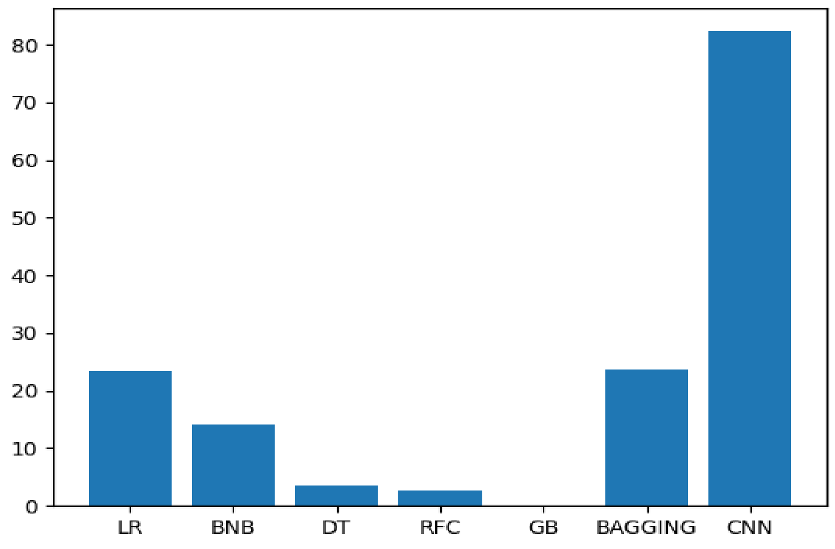


Fig17. Loss comparison of the algorithms.

CONCLUSION

In this paper we have experimented various a Machine learning algorithm such as such as Random Forest, Decision Tree, Gradient Boosting, Bagging Classifier, Logistic Regression, Bernoulli-NB to predict the software quality tested with two datasets. The research outcome of proposed model has increased the values of accuracy, precision, recall and f-measure when compared with other models. Using CMMI we achieve good quality software done by ISO 9000/9001-3 computer software which is organized into goals, commitments, abilities etc. In future, we plan to apply more evolutionary and optimization techniques for classification to compare Various algorithm using data sets.

REFERENCES

1. https://www.geeksforgeeks.org/image-classifier-using-cnn/.
2. https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/.
3. https://towardsdatascience.com/the-4-convolutional-neural-network-models-that-can-classify-your-fashion-images-9fe7f3e5399d.
4. Chayanika Sharma, Sangeeta Sabharwal, Ritu Sibal, "A Survey on Software Testing Techniques using Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1, January 2013.
5. 1Faisal Shafique Butt, Sundus Shaukat, Wasif Nisar, Ehsan Ullah Munir, "Software Quality Assurance in Software Projects: A Study of Pakistan", Research Journal of Applied Sciences, Engineering and Technology 5(18): 4568-4575, 2013.
6. Akshat Sharma, Rishon Patani, Ashish Aggarwal, "Software Testing Using Genetic Algorithms", International Journal of Computer Science & Engineering Survey (IJCSES) Vol.7, No.2, April 2016.
7. A. Ayberk CERAN, Ö. Özgür TANRIÖVER, "An experimental study for software quality prediction with machine learning methods", University of Vermont Libraries9, 78-1-7281-9352-6/20/$31.00 ©2020 IEEE, august 2020.
8. Muni Dhanalakshmi Kumbakonam, Mahammad Shafi RajaSaheb, "measuring application performance and load testing with scenario schedules", IJIRAE, Issue 09, Volume 7 (September 2020)
9. E.J. Weyuker, "On Testing Non-Testable Programs", Computer Journal vol.25 no.4, November 1982, pp.465-470.

10. P. Long and R. Servedio, “Martingale Boosting”, Eighteenth Annual Conference on Computational Learning Theory (COLT), Bertinoro, Italy, 2005, pp. 79-94.
11. E. Walton, “Data Generation for Machine Learning Techniques”, [http://www.cs.bris.ac.uk/Teaching/Resources/COMS30500/ ExampleTheses/thesis6.pdf], University of Bristol, 2001.
12. Bawane, N. and C. Srikrishna, 2010. A novel method for quantitative assessment of software quality. Int. J. Comput. Sci. Secur., 3(6): 508.
13. Farooq, S.U., S. Quadri and N. Ahmad, 2011. Software measurements and metrics: Role in effective software testing. Int. J. Eng. Sci. Technol., 3(1): 671-680.
14. D. Bowes, T. Hall, and J. Petrić "Software defect prediction: do different classifiers find the same defects", Software Quality Journal, 26(2), 2018, pp. 525-552.
15. E. Rashid, S. Patnaik, and V. Bhattacherjee, "Software quality estimation using machine learning: Case-Based reasoning technique, " International Journal of Computer Applications, 2012.
16. Dyana Rashid Ibrahim, Rawan Ghnemat, Amjad Hudaib, “Software Defect Prediction using Feature Selection and Random Forest Algorithm”, 2017 International Conference on New Trends in Computing Sciences.
17. Shaik Hazar Babavali, K.Rahul, N.V. Vishnu Teja, E. Sri Devi, “Software Defect Detection Using Decision Tree Learning Algorithms”, Journal of Xi'an University of Architecture & Technology, ISSN No : 1006-7930, Volume XII, Issue IV, 2020.
18. Munidhanalakshmi Kumbakonam and Mahammad Shafi RajaSaheb “MEASURING APPLICATION PERFORMANCE AND LOAD TESTING WITH SCENARIO SCHEDULES”. International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163 Issue 09, Volume 7 (September 2020).
19. Munidhanalakshmi Kumbakonam and Mahammad Shafi RajaSaheb “Correlation for Dynamic Values Using WDiff Performance Testing Technique”. International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-7, Issue-2, February2021 ISSN: 2395-3470.
20. Pavan Madduru "Artificial Intelligence as A Service in Distributed Multi Access Edge Computing On 5g Extracting Data Using IOT And Including AR/VR For Real-Time Reporting". IT in Industry, Vol. 9, No.1, 2021, Published Online 15-March-2021.