IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

AN EXPERIMENTAL STUDY FOR SOFTWARE QUALITY PREDICTION USING MACHINE LEARNING METHODS

1Regulapati Akhila, 2A.Anusha, 3Dr.P.Srinivasa Rao

1scholar Of Computer Science and Engineering, 2Assistant Professor, 3Hod and Associate Professor

1JB Institute of Engineering and Technology,

2JB Institute of Engineering and Technology,

3JB Institute of Engineering and Technology

ABSTRACT

Keywords: — Estimation, Machine Learning, Software Quality..

Software quality estimation is an activity needed at various stages of software development. It may be used for planning the project`s quality assurance practices and for benchmarking. In earlier previous studies, two methods (Multiple Criteria Linear Programming and Multiple Criteria Quadratic Programming) for estimating the quality of software had been used. Also, C5.0, SVM and Neutral network were experimented with for quality estimation. These studies have relatively low accuracies. In this study, we aimed to improve estimation accuracy by using relevant features of a large dataset. We used a feature selection method and correlation matrix for reaching higher accuracies. In addition, we have experimented with recent methods shown to be successful for other prediction tasks. Machine learning algorithms such as Xgboost, Random Forest and Decision Tree are applied to the data to predict the software quality and reveal the relation between the quality and development attributes. The experimental results show that the quality level of software can be well estimated by machine learning algorithms.

1. INTRODUCTION

1.1 Introduction

Software applications may contain defects, originating from requirements analysis, specification and other activities conducted in the software development. Therefore, software quality estimation is an activity needed at various stages [1]. It may be used for planning the project based quality assurance practices and for benchmarking. In addition, the number of defects per unit is considered one of the most important factors that indicate the quality of the software [2].

There are two directly comparable studies on software quality prediction using defect quantities in ISBGS dataset. In the first study, the two methods (MCLP and MCQP) were experimented with the dataset and the results were compared [3]. The quality level was classified according to: number of minor defect + 2*number of major defect + 4*number of extreme defect. The quality of level was to be either high or low. They used k-fold cross-validation technique to measure MCLP and MCQP's performance on the ISBSG database. Release 10 Dataset (released in January 2007) which contained 4,017 records and 106 attributes was used. After preprocessing, 374 records and 11 attributes remained in the dataset.

In another study, the same data set was used again [4]. The software belonged to high quality class if it fulfills the following requirements: the extreme defects exist or the number of major defects is more than 1 or the number of minor defects is more than 10. The rest are assumed to belong to low quality class. After preprocessing, 746 projects and 53 attributes remained in the dataset. They used C5.0, SVM and Neutral network for classification.

As an example to a more application oriented study Rashid et al. [5] used case based reasoning (CBR) for software quality estimation. CBR is a machine learning model which performs the learning process using the results of the previous experiments. Line of code, number of function, difficulty level, and development type and programmers experience are entered and these attributes are used for estimation. The deviation is calculated by using Euclidian distance (ED) or The Manhattan distance (MD). If the error in estimation is less than 10% then the record is saved to the database. Number of inputs that can be obtained from the user is limited. Also, it is necessary to have close values in the database in order to estimating precise values.

In these studies, quality estimation was done by binary classification. We tried to improve these prediction models, taking into account the size in terms of function points and using 4-level classification. We have experimented with recent classification methods shown to be successful for other prediction tasks.

1.2 Existing System

Software quality estimation is an activity needed at various stages of software development. It may be used for planning the project`s quality assurance practices and for benchmarking. In earlier previous studies, two methods (Multiple Criteria Linear Programming and Multiple Criteria Quadratic Programming) for estimating the quality of software had been used. Also, C5.0, SVM and Neutral network were experimented with for quality estimation. These studies have relatively low accuracies.

1.2.1 Demerits of Existing System

- To improve estimation accuracy by using relevant features of a large dataset.

1.3 Proposed System

In this paper We used a feature selection method and correlation matrix for reaching higher accuracies. In addition, we have experimented with recent methods shown to be successful for other prediction tasks. Machine learning algorithms such as Xgboost, Random Forest and Decision Tree are applied to the data to predict the software quality and reveal the relation between the quality and development attributes. The experimental results show that the quality level of software can be well estimated by machine learning algorithms.

1.3.1 Merits of Proposed System

2. Improved estimation accuracy by using relevant features of a large dataset.

Literature Survey

[1] Vijay, T. John, D. M. G. Chand, and D. H. Done. "Software quality metrics in quality assurance to study the impact of external factors related to time." International Journal of Advanced Research in Computer Science and Software Engineering, 2017.

The purpose of this research is to build a software quality assessment model to evaluate the quality of mobile-based elderly fall detection software. The assessment is based on the quality factors found in the software quality model. The quality factor is adjusted to the characteristics of the software. The model is needed because the software has its own characteristics. This research consists of several stages. The first thing to do is analysing the software domain to determine its characteristics. The second is defining the software assessment needs by mapping software characteristics with the quality standards used (ISO / IEC 25010: 2011) to obtain the appropriate quality factors. The software quality metrics is determined after the quality factors is obtained. The metric to be used is Goal Question Metrics (GCM). The third is software quality weighting process, including its criterias and sub-criterias. Determination of the equation for software quality assesment is the final stage of the research. Based on the reseach process, it can be concluded that the model developed successfully can be used to assess the software.

[2] 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.

During the last 10 years, hundreds of different defect prediction models have been published. The performance of the classifiers used in these models is reported to be similar with models rarely performing above the predictive performance ceiling of about 80% recall. We investigate the individual defects that four classifiers predict and analyse the level of prediction uncertainty produced by these classifiers. We perform a sensitivity analysis to compare the performance of Random Forest, Naïve Bayes, RPart and SVM classifiers when predicting defects in NASA, open source and commercial datasets. The defect predictions that each classifier makes is captured in a confusion matrix and the prediction uncertainty of each classifier is compared. Despite similar predictive performance values for these four classifiers, each detects different sets of defects. Some classifiers are more consistent in predicting defects than others. Our results confirm that a unique subset of defects can be detected by specific classifiers. However, while some classifiers are consistent in the predictions they make, other classifiers vary in their predictions. Given our results, we conclude that classifier ensembles with decision-making strategies not based on majority voting are likely to perform best in defect prediction.

2.1 Requirements Specifications

2.1.1 Hardware Requirements:

•	System	:	Intell I-3, 5, 7 Processor.
•	Hard Disk	:	500 GB.
•	Floppy Drive	:	1.44 Mb.
•	Monitor	:	14’ Colour Monitor.
•	Mouse	:	Optical Mouse.
•	Ram	:	2Gb.

2.1.2 Software Requirements:

•	Operating system	**:**	Windows 7,8,10 Ultimate, Linux, Mac.
•	Front-End	**:**	Python.
•	Coding Language	**:**	Python.
•	Software Environment	**:**	Anaconda(jupyter or spyder).

2.1.3 User Requirements

- User has to load the application before using it
- User needs to have data (text, audio, image, video) which is to be hidden
- User needs to have a master file in which he/she wants to hide the data
- User needs to have a stego-key in order to encrypt or decrypt the data

2.1.4 Functional Requirements

Outputs from computer systems are required primarily to communicate the results of processingto users. They are also used to provide a permanent copy of the results for later consultation.

The various types of outputs in general are:

External Outputs, whose destination is outside the organization,.

- Internal Outputs whose destination is within organization and they are theuser's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.
- Understanding user's preferences, expertise level and his business requirements through a friendly questionnaire.
- Input data can be in four different forms - Relational DB, text files, .xls and xml files. For testing and demo you can choose data from any domain. User-B can provide business dataas input.

2.2 System Study

2.2.1 Feasibility Study

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility
study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility

- Economical Feasibility

The main objective of the study of the prospect is just to verify the technical, social as well as operational achievement to include additional features and to investigate the past working structure. Every device will be back to earth in the event that they are infinite energy and unremitting time. Perspectives exist in the concentration of attainability of some portion of the quick evaluation.

2.2.2 Economic Feasibility

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economicalfeasibility for certain.

2.2.3 Feasibility of operation

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken asan important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.2.4 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the datarequired to use the new system?
- Will the proposed system provide adequate response to inquiries, regardlessof the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’.

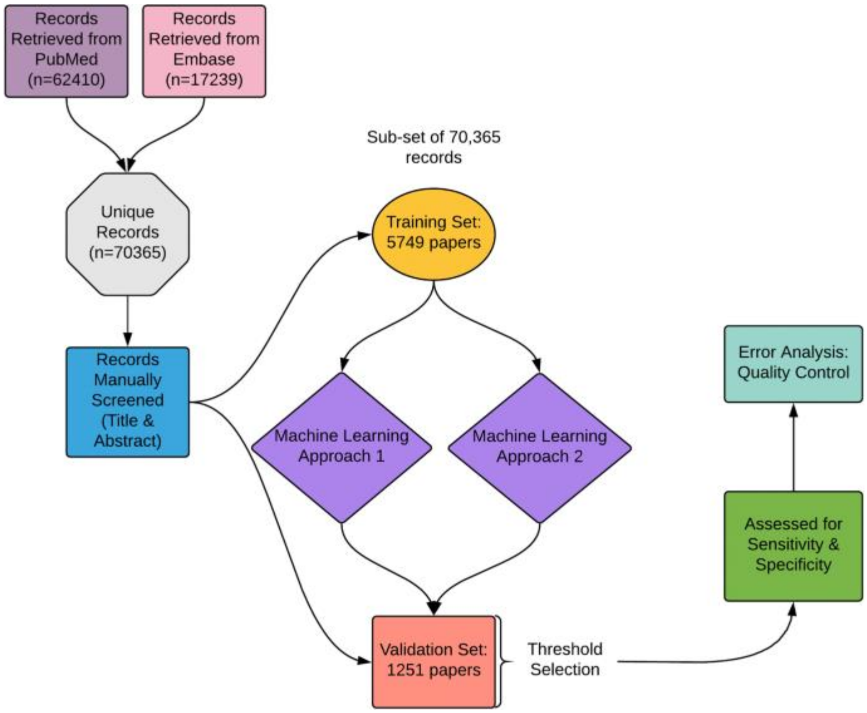
The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose isto create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providinga fast feedback to the users irrespective of the number of users using the system.

3. SYSTEM DESIGN

Data flow diagrams delineate the way the data is stored within a system as well as input and return resources are concerned. It is easy to use information source diagrams to consider giving some market function. The strategy begins with an introductory image of the company and progresses by disintegrating each of the beneficial unmistakably outposts.

3.1 System Architecture

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first stepin the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Beginning, once systemrequirement have been specified and analyzed, system design is the first of the three technicalactivities -design, code and test that is required to build and verify software.


Records Retrieved from PubMed (n=62410)
Records Retrieved from Embase (n=17239)
Unique Records (n=70365)
Records Manually Screened (Title & Abstract)
Sub-set of 70,365 records
Training Set: 5749 papers
Machine Learning Approach 1
Machine Learning Approach 2
Validation Set: 1251 papers
Threshold Selection
Assessed for Sensitivity & Specificity
Error Analysis: Quality Control

3.2 Uml Diagrams

3.2.1 Data Flow Diagram:

The Data Flow Diagram (DFD) is used to display parts of the system. These components are the system process, the details the process uses, an external dimension that interacts within the system with the frame and the digital data. DFD demonstrates how the information goes through the system and how it is balanced by a movement of changes. It is a pictorial framework that represents the distribution of information and the progressions introduced as motions of data from engagement to yield. A DFD may be used at any level of consideration for addressing a structure.

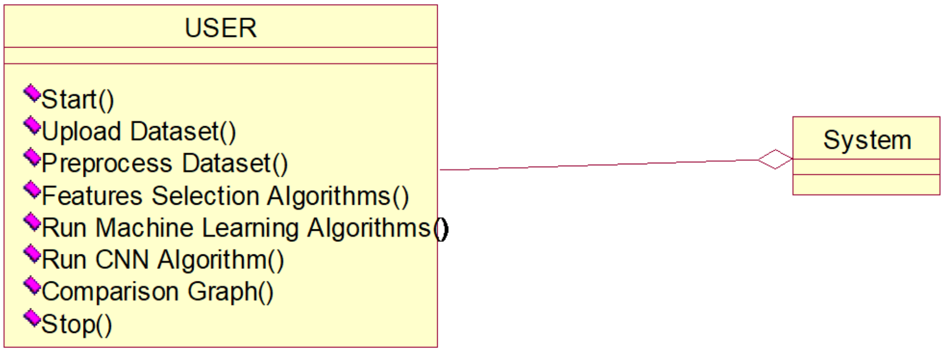
3.2.2 Class diagram:

The class outline is the essential structure square of occasion driven information investigation. It is utilized both for by and large sensible confirmation of the application's exactness and for detailed show making a model translation into software code. Equally, class graphs can be used to demonstrate information.

Every class in the class chart resembles both to the fundamental items, to the application participations and to the classes to be altered. A class has three locales; classes with boxes which contain three sections are addressed in the framework:

- The name is given by the first / top component
- The pivot section includes the class features
- the techniques or actions which the class may take or throw back is shown at the base

The underneath Fig 3.3 shows the class diagram of the project



3.2.3 Use case diagram:

A use case at any rate troublesome is a representation of a customer's correspondence with the structure and outlines the conclusions of a use case. A usage case analysis will delineate the various forms of a structure's users and the different ways they help out the framework. Typically this type of graph is used identified with the stamped use case but will also be regularly merged by different types of layouts. The underneath Fig 3.4 shows the utilization case diagram of the task.

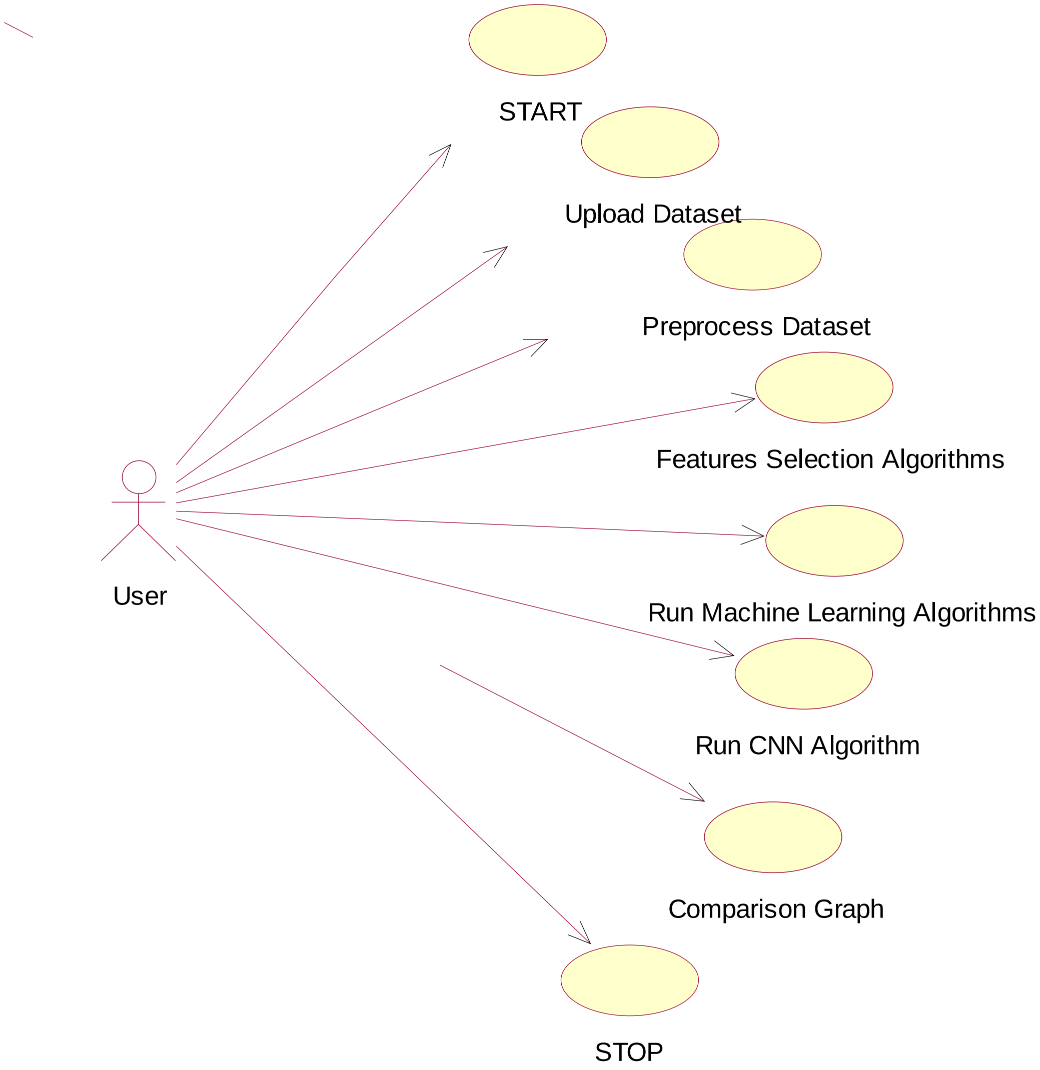
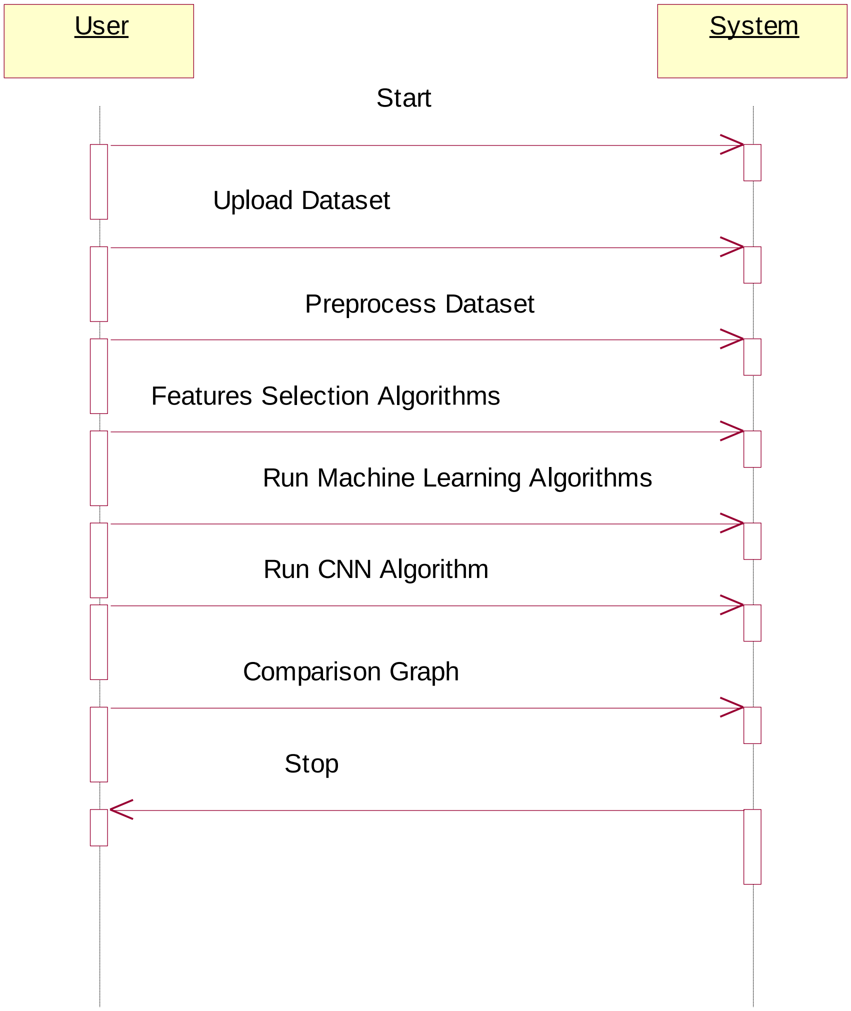


Fig 3.4: Use case Diagram

3.2.4 Sequence Diagram:

The relationship description that demonstrates how, and under what case, the procedures operate together is said to be sequence graph. A grouping graph shows orchestrated connections of objects in succession of time. It depicts the objects and classes associated with either the circumstance, as well as the grouping of signals exchanged in between documents assumed to make the whole thing suitable. Sequences diagrams are usually related to usage case recognition of a function in progress in the Functional View of the system. Succession shapes are occasionally called graphs of the occasion, occasional situations, and timing charts. The beneath Fig 3.5 and Fig 3.6 shows the sequence graphs of the encryption and decryption process individually

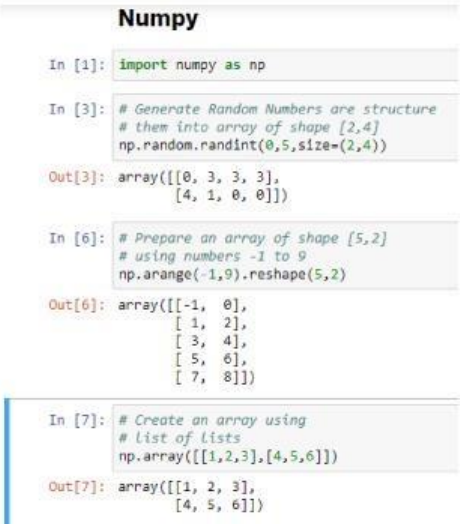

User
System
Start
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Stop

4. METHODOLOGY & IMPLEMENTATION

4.1 Modules:

1. Numpy

Python has a strong set of data types and data structures. Yet it wasn't designed for Machine Learning per say. Enter numpy (pronounced as num-pee). Numpy is a data handling library, particularly one which allows us to handle large multi-dimensional arrays along with a huge collection of mathematical operations. The following is a quick snippet of numpy in action.



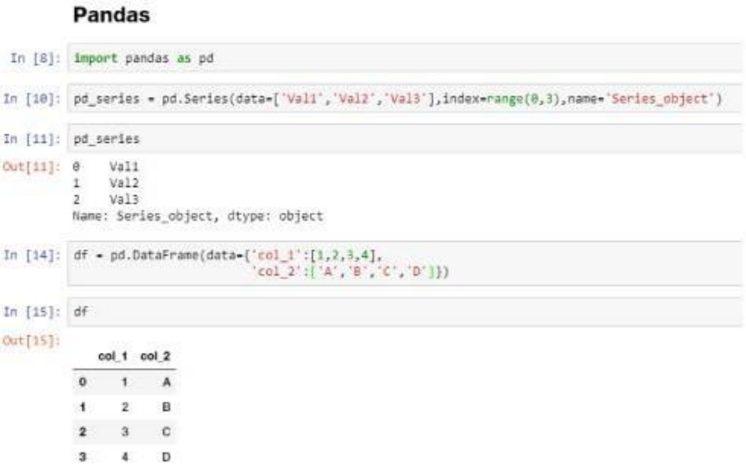
Numpy

```
In [1]: import numpy as np

In [3]: # Generate Random Numbers are structure
        # them into array of shape [2,4]
        np.random.randint(0,5,size=(2,4))

Out[3]: array([[0, 3, 3, 3],
               [4, 1, 0, 0]])

In [6]: # Prepare an array of shape [5,2]
        # using numbers -1 to 9
        np.arange(-1,9).reshape(5,2)

Out[6]: array([[-1,  0],
               [ 1,  2],
               [ 3,  4],
               [ 5,  6],
               [ 7,  8]])

In [7]: # Create an array using
        # list of lists
        np.array([[1,2,3],[4,5,6]])

Out[7]: array([[1, 2, 3],
               [4, 5, 6]])
```

Numpy isn't just a data handling library known for its capability to handle multidimensional data. It is also known for its speed of execution and vectorization capabilities. It provides MATLAB style functionality and hence requires some learning before you can get comfortable. It is also a core dependency for other majorly used libraries like pandas, matplotlib and so on.It's documentation itself is a good starting point.

2. Pandas

Think of relational data, think pandas. Yes, pandas is a python library that provides flexible and expressive data structures (like dataframes and series) for data manipulation. Built on top of numpy, pandas is as fast and yet easier to use.



Pandas

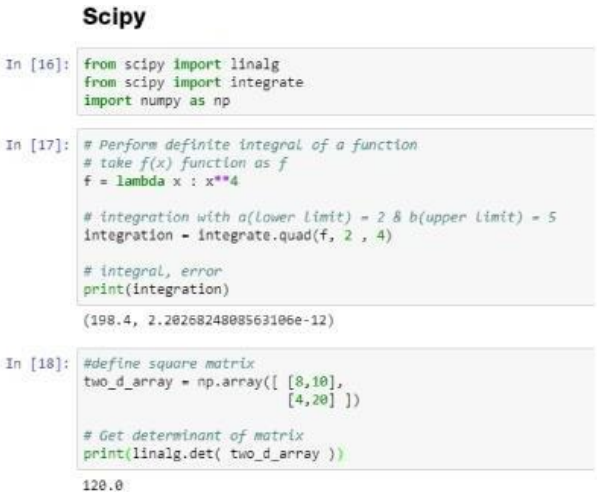
```
In [8]: import pandas as pd

In [10]: pd_series = pd.Series(data=['Val1','Val2','Val3'],index=range(0,3),name='Series_object')

In [11]: pd_series

Out[11]: 0    Val1
         1    Val2
         2    Val3
         Name: Series_object, dtype: object

In [14]: df = pd.DataFrame(data={'col_1':[1,2,3,4],
                                 'col_2':['A','B','C','D']})

In [15]: df

Out[15]:
```

	col_1	col_2
0	1	A
1	2	B
2	3	C
3	4	D

Pandas provides capabilities to read and write data from different sources like CSVs, Excel, SQL Databases, HDFS and many more. It provides functionality to add, update and delete columns, combine or split dataframes/series, handle datetime objects, impute null/missing values, handle time series data, conversion to and from numpy objects and so on. If you are working on a real-world Machine Learning use case, chances are, you would need pandas sooner than later. Similar to numpy, pandas is also an important component of the SciPy or Scientific Python Stack .

3. Scipy

Pronounced as Sigh-Pie, this is one of the most important python libraries of all time. Scipy is a scientific computing library for python. It is also built on top of numpy and is a part of the Scipy Stack.



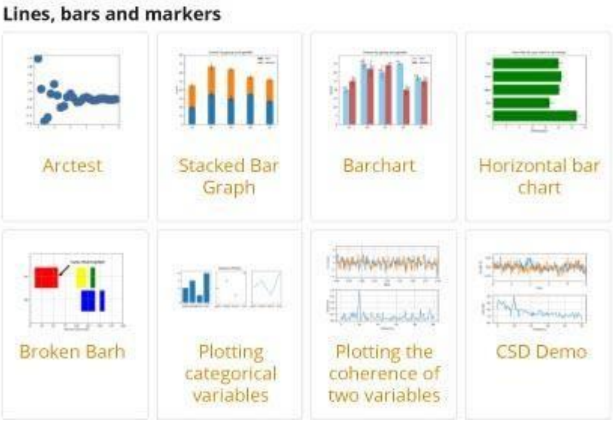
Scipy

```
In [16]: from scipy import linalg
         from scipy import integrate
         import numpy as np

In [17]: # Perform definite integral of a function
         # take f(x) function as f
         f = lambda x : x**4

         # integration with a(lower limit) = 2 & b(upper limit) = 5
         integration = integrate.quad(f, 2 , 4)

         # integral, error
         print(integration)

         (198.4, 2.2026824808563106e-12)

In [18]: #define square matrix
         two_d_array = np.array([ [8,10],
                                  [4,20] ])

         # Get determinant of matrix
         print(linalg.det( two_d_array ))

         120.0
```

This is yet another *behind the scenes* library which does a whole lot of heavy lifting. It provides modules/algorithms for linear algebra, integration, image processing, optimizations, clustering, sparse matrix manipulation and many more. .

4 . Matplotlib

Another component of the SciPy stack, matplotlib is essentially a visualization library. It works seamlessly with numpy objects (and its high-level derivatives like pandas). Matplotlib provides a MATLAB like plotting environment to prepare high-quality figures/charts for publications, notebooks, web applications and so on.

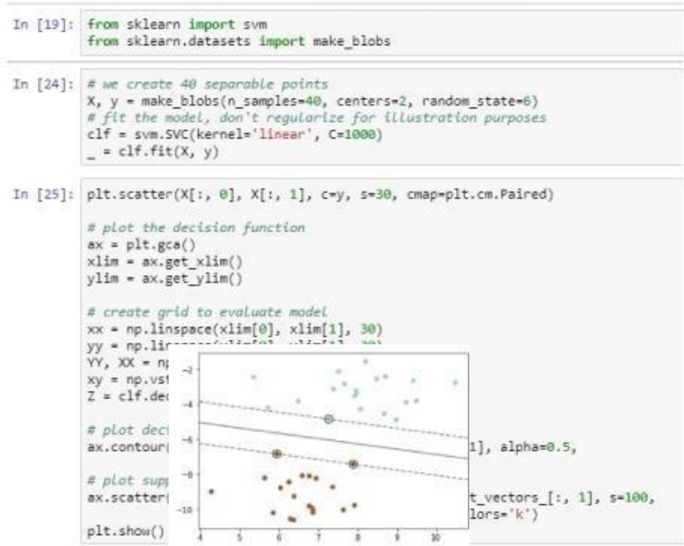


Matplolib is a high customizable low-level library that provides a whole lot of controls and knobs to prepare any type of visualization/figure. Given its low-level nature, it requires a bit of getting used to along with plenty of code to get stuff done. Its well documented and extensible design has allowed a whole list of high-level visualization libraries to be built on top. Some of which, we will discuss in the coming sections. :

5 . Scikit-Learn

Designed as an extension to the SciPy library, scikit-learn has become the de-facto standard for many of the machine learning tasks. Developed as part of Google Summer of Code project,it has now become a widely contributed o

Scikit-Learn {Source: Sklearn Examples}



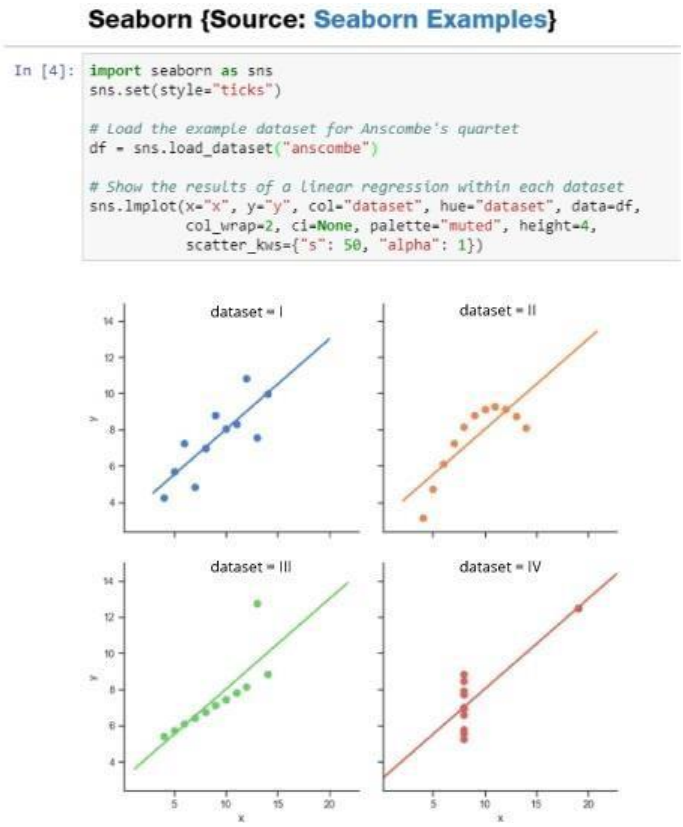
```
In [19]: from sklearn import svm
         from sklearn.datasets import make_blobs

In [24]: # we create 40 separable points
         X, y = make_blobs(n_samples=40, centers=2, random_state=6)
         # fit the model, don't regularize for illustration purposes
         clf = svm.SVC(kernel='linear', C=1000)
         _ = clf.fit(X, y)

In [25]: plt.scatter(X[:, 0], X[:, 1], c=y, s=30, cmap=plt.cm.Paired)

         # plot the decision function
         ax = plt.gca()
         xlim = ax.get_xlim()
         ylim = ax.get_ylim()

         # create grid to evaluate model
         xx = np.linspace(xlim[0], xlim[1], 30)
         yy = np.lir
         YY, XX = nr
         xy = np.vst
         Z = clf.dec

         # plot deci
         ax.contour(                                   1], alpha=0.5,

         # plot supr
         ax.scatter(                                   t_vectors_[:, 1], s=100,
                                                       lors='k')

         plt.show()
```

Scikit-learn provides a simple yet powerful fit-transform and predict paradigm to learn from data, transform the data and finally predict. Using this interface, it provides capabilities to prepare classification, regression, clustering and ensemble models. It also provides a multitude of utilities for preprocessing, metrics, model evaluation techniques, etc.

Visualization

6. Seaborn

Built on top of matplotlib, seaborn is a high-level visualization library. It provides sophisticated styles straight out of the box (which would take some good amount of effort if done using matplotlib).



Seaborn {Source: Seaborn Examples}

```
In [4]: import seaborn as sns
        sns.set(style="ticks")

        # Load the example dataset for Anscombe's quartet
        df = sns.load_dataset("anscombe")

        # Show the results of a linear regression within each dataset
        sns.lmplot(x="x", y="y", col="dataset", hue="dataset", data=df,
                   col_wrap=2, ci=None, palette="muted", height=4,
                   scatter_kws={"s": 50, "alpha": 1})
```

Sample plots using seaborn.

Apart from styling prowess and sophisticated color pallets, seaborn provides a range of visualizations and capabilities to work with multivariate analysis. It provides capabilities to perform regression analysis, handling of categorical variables and aggregate statistics.

4.2 INPUT DESIGN

Input design is a part of overall system design. The main objective during the inputdesign is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

INPUT STAGES:

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation
- Data correction

INPUT TYPES:

It is necessary to determine the various types of inputs. Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system.
- Internal inputs, which are user communications with the system.
- Operational, which are computer department’s communications to the system?
- Interactive, which are inputs entered during a dialogue.

INPUT MEDIA:

At this stage choice has to be made about the input media. To conclude about the inputmedia consideration has to be given to;

- Type of input
- Flexibility of format
- Speed
- Accuracy
- Verification methods
- Rejection rates
- Ease of correction
- Storage and handling requirements
- Security
- Easy to use
- Portability

Keeping in view the above description of the input types and input media, it can be saidthat most of the inputs are of the form of internal and interactive. As

Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

4.3 OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization
- Internal Outputs whose destination is within organization and they are the
- User’s main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly.

OUTPUT DEFINITION

The outputs should be defined in terms of the following points:

- Type of the output
- Content of the output
- Format of the output
- Location of the output
- Frequency of the output
- Volume of the output
- Sequence of the output

It is not always desirable to print or display data as it is held on a computer. It shouldbe decided as which form of the output is the most suitable.

4.4 CODE:

```
   fromtkinter import messagebox
fromtkinter import *
fromtkinter import simpledialog
importtkinter
fromtkinter import filedialog
importmatplotlib.pyplot as plt
fromtkinter.filedialog import askopenfilename
fromsklearn.model_selection import train_test_split
fromsklearn.metrics import accuracy_score
importnumpy as np
import pandas as pd
fromgenetic_selection import GeneticSelectionCV
fromsklearn.metrics import classification_report
fromsklearn.metrics import confusion_matrix
fromsklearn import svm
```

```
fromkeras.models import Sequential
fromkeras.layers import Dense
import time

main = tkinter.Tk()
main.title("Android Malware Detection")
main.geometry("1300x1200")

global filename
global train
globalsvm_acc, nn_acc, svmga_acc, annga_acc
globalX_train, X_test, y_train, y_test
globalsvmga_classifier
globalnnga_classifier
globalsvm_time,svmga_time,nn_time,nnga_time



def upload():
global filename
filename = filedialog.askopenfilename(initialdir="dataset")
pathlabel.config(text=filename)
text.delete('1.0', END)
text.insert(END,filename+" loaded\n");

defgenerateModel():
globalX_train, X_test, y_train, y_test
text.delete('1.0', END)
```

```
train = pd.read_csv(filename)
rows = train.shape[0]  # gives number of row count
cols = train.shape[1]  # gives number of col count
features = cols - 1
print(features)
    X = train.values[:, 0:features]
    Y = train.values[:, features]
print(Y)
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2,
random_state = 0)


text.insert(END,"Dataset Length : "+str(len(X))+"\n");
text.insert(END,"Splitted Training Length : "+str(len(X_train))+"\n");
text.insert(END,"Splitted Test Length : "+str(len(X_test))+"\n\n");


def prediction(X_test, cls):  #prediction done here
y_pred = cls.predict(X_test)
fori in range(len(X_test)):
print("X=%s, Predicted=%s" % (X_test[i], y_pred[i]))
returny_pred

# Function to calculate accuracy
defcal_accuracy(y_test, y_pred, details):
    cm = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test,y_pred)*100
```

```
text.insert(END,details+"\n\n")
text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")
text.insert(END,"Report : "+str(classification_report(y_test, y_pred))+"\n")
text.insert(END,"Confusion Matrix : "+str(cm)+"\n\n\n\n\n")
return accuracy

defrunSVM():
globalsvm_acc
globalsvm_time
start_time = time.time()
text.delete('1.0', END)
cls = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random_state = 2)
cls.fit(X_train, y_train)
prediction_data = prediction(X_test, cls)
svm_acc = cal_accuracy(y_test, prediction_data,'SVM Accuracy')
svm_time = (time.time() - start_time)

defrunSVMGenetic():
text.delete('1.0', END)
globalsvmga_acc
globalsvmga_classifier
globalsvmga_time
estimator = svm.SVC(C=2.0,gamma='scale',kernel = 'rbf', random_state = 2)
svmga_classifier = GeneticSelectionCV(estimator,
cv=5,
verbose=1,
scoring="accuracy",
max_features=5,
n_population=50,
crossover_proba=0.5,
mutation_proba=0.2,
n_generations=40,
crossover_independent_proba=0.5,
mutation_independent_proba=0.05,
tournament_size=3,
n_gen_no_change=10,
caching=True,
n_jobs=-1)
start_time = time.time()
svmga_classifier = svmga_classifier.fit(X_train, y_train)
svmga_time = svm_time/2
prediction_data = prediction(X_test, svmga_classifier)
svmga_acc = cal_accuracy(y_test, prediction_data,'SVM with GA Algorithm
Accuracy, Classification Report & Confusion Matrix')


defrunNN():
globalnn_acc
globalnn_time
text.delete('1.0', END)
start_time = time.time()
model = Sequential()
model.add(Dense(4, input_dim=215, activation='relu'))
model.add(Dense(215, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
model.fit(X_train, y_train, epochs=50, batch_size=64)
    _, ann_acc = model.evaluate(X_test, y_test)
nn_acc = ann_acc*100
text.insert(END,"ANN Accuracy : "+str(nn_acc)+"\n\n")
nn_time = (time.time() - start_time)

defrunNNGenetic():
globalannga_acc
globalnnga_time
text.delete('1.0', END)
train = pd.read_csv(filename)
rows = train.shape[0]  # gives number of row count
cols = train.shape[1]  # gives number of col count
features = cols - 1
print(features)
    X = train.values[:, 0:100]
    Y = train.values[:, features]
print(Y)
    X_train1, X_test1, y_train1, y_test1 = train_test_split(X, Y, test_size = 0.2,
random_state = 0)
model = Sequential()
model.add(Dense(4, input_dim=100, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])
start_time = time.time()
model.fit(X_train1, y_train1)
nnga_time = (time.time() - start_time)
   _, ann_acc = model.evaluate(X_test1, y_test1)
annga_acc = ann_acc*100
text.insert(END,"ANN with Genetic Algorithm Accuracy :
"+str(annga_acc)+"\n\n")

def graph():
height = [svm_acc, nn_acc, svmga_acc, annga_acc]
   bars = ('SVM Accuracy','NNAccuracy','SVM Genetic Acc','NN Genetic Acc')
y_pos = np.arange(len(bars))
plt.bar(y_pos, height)
plt.xticks(y_pos, bars)
plt.show()



deftimeGraph():
height = [svm_time,svmga_time,nn_time,nnga_time]
   bars = ('SVM Time','SVM Genetic Time','NNTime','NN Genetic Time')
y_pos = np.arange(len(bars))
plt.bar(y_pos, height)
plt.xticks(y_pos, bars)
plt.show()
```

```
font = ('times', 16, 'bold')
title = Label(main, text='Android Malware Detection Using Genetic Algorithm
based Optimized Feature Selection and Machine Learning')
#title.config(bg='brown', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 14, 'bold')
uploadButton = Button(main, text="Upload Android Malware Dataset",
command=upload)
uploadButton.place(x=50,y=100)
uploadButton.config(font=font1)

pathlabel = Label(main)
pathlabel.config(bg='brown', fg='white')
pathlabel.config(font=font1)
pathlabel.place(x=460,y=100)

generateButton = Button(main, text="Generate Train & Test Model",
command=generateModel)
generateButton.place(x=50,y=150)
generateButton.config(font=font1)

svmButton = Button(main, text="Run SVM Algorithm", command=runSVM)
svmButton.place(x=330,y=150)
svmButton.config(font=font1)

svmgaButton = Button(main, text="Run SVM with Genetic Algorithm",
command=runSVMGenetic)
svmgaButton.place(x=540,y=150)
svmgaButton.config(font=font1)

nnButton = Button(main, text="Run Neural Network Algorithm",
command=runNN)
nnButton.place(x=870,y=150)
nnButton.config(font=font1)

nngaButton = Button(main, text="Run Neural Network with Genetic Algorithm",
command=runNNGenetic)
nngaButton.place(x=50,y=200)
nngaButton.config(font=font1)

graphButton = Button(main, text="Accuracy Graph", command=graph)
graphButton.place(x=460,y=200)
graphButton.config(font=font1)

exitButton = Button(main, text="Execution Time Graph", command=timeGraph)
exitButton.place(x=650,y=200)
exitButton.config(font=font1)



font1 = ('times', 12, 'bold')
```

```
text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=10,y=250)

text.config(font=font1)

#main.config()

main.mainloop()
```

5. Result Analysis

5.1 Testing

Testing also known to be validation is the place where the test data is masterminded and used for independent analysis of the modules and the endorsement given for the fields later. By then the inspection of the framework occurs which ensures that all parts of the system fill in as a unit appropriately. The test data must be assigned with the ultimate goal that it experienced any condition possible. Next comes the delineation of the test methods that were performed during the test period.

5.1.1 System Testing

Screening/Testing is now an important part of any system or mission, especially in the field of technology advancement. The importance of testing is a technique towards legitimizing, throughout the situation that one should be equipped to go forward, that it is to be tested whether one is in a position to hold up to the challenges of a specific situation, and this is the justification for checking before progress is achieved. At When designing the product before it is offered to the user to buy the product, it must be checked if it answers the purpose it is made. This measure involves various kinds from which the commodity can be assured good. The program was smartly tried and instance of program execution is revisited for a plethora of information. Consequently, the code was thoroughly inspected to most thinkable accurate data and along these the conclusions were evaluated.

5.1.2 Module Testing

Per module is tested independently to discover the blunders. This enables us to identify and correct mistakes without affecting other modules. Wherever the program does not meet the necessary efficiency, then it should be revised in order to obtain the necessary result. All modules are thus tested separately from base to start with either the smallest or the least modules and continue to the next level. The whole subsystem is evaluated with different workers and their harsh processing time, and the test's outcome is compared with the tests that are formally organized. Per module is independently tried inside the system. The investment portfolio order and job preparation modules are independently evaluated in this context and their relevant outcomes are obtained that reduces the time-holding process.

5.1.3 Integration Testing

The reconciliation test is applied after testing of the module. These blunders are balanced when doing this checking when linking the modules there can be a risk that errors occur. Both modules are linked and tested out in this setting. The test results are outstandingly correct. Accordingly, the methodology accurately maps jobs and careers with assets.

5.1.4 Acceptance Testing

So at moment where that client did not finish any serious problems with its accuracy, the model passes out through final recognition test. Such an assessment confirms that perhaps the fixed support the very first objectives, goals and requirements generated during the investigation without proper implementation leads to time wastage and cash assurance evaluations on the neck of customers and the board.

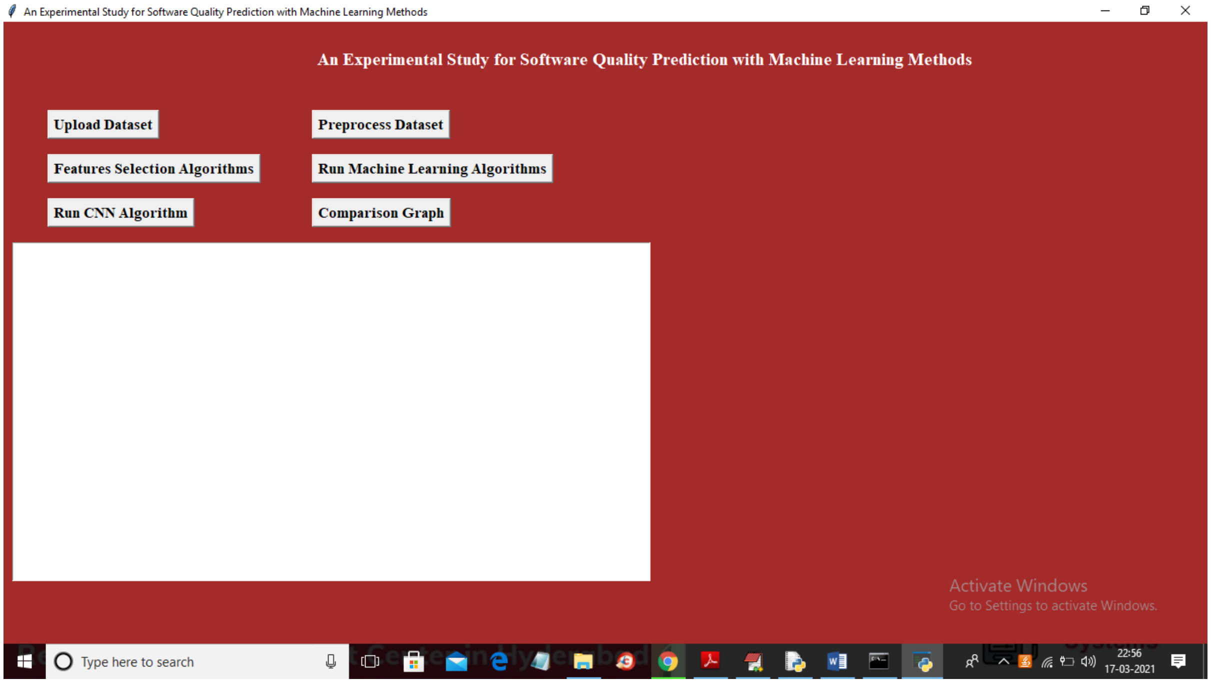
5.1.5 Test Results

Every case study referenced above passed effectively. No deformities experienced.

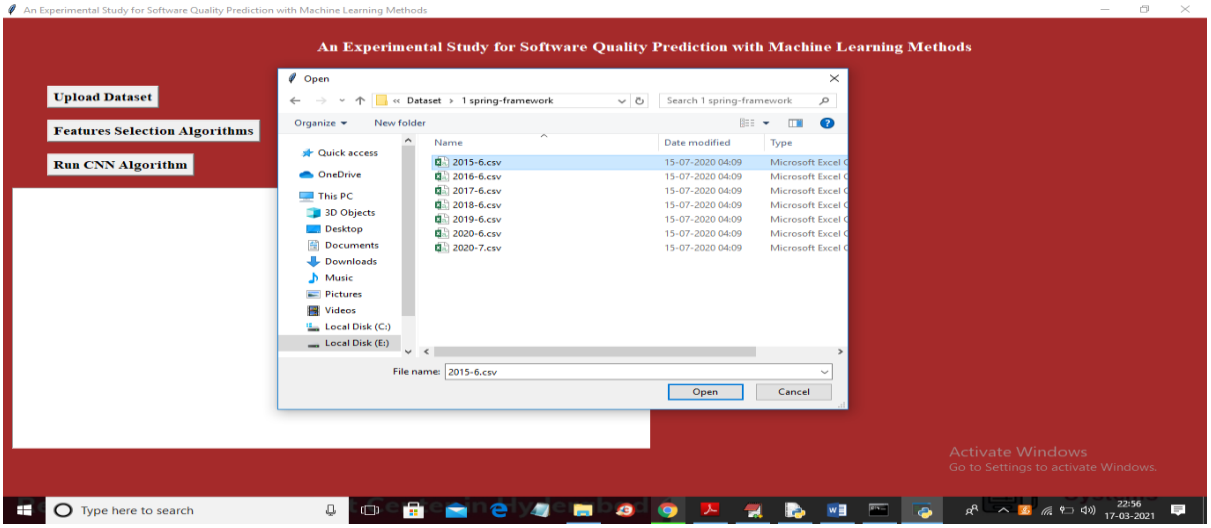
5.2 Experimental Outcomes

5.2.1 Output Screens

To run project double click on ‘run.bat’ file 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
Activate Windows
Go to Settings to activate Windows.
Type here to search
22:56
17-03-2021

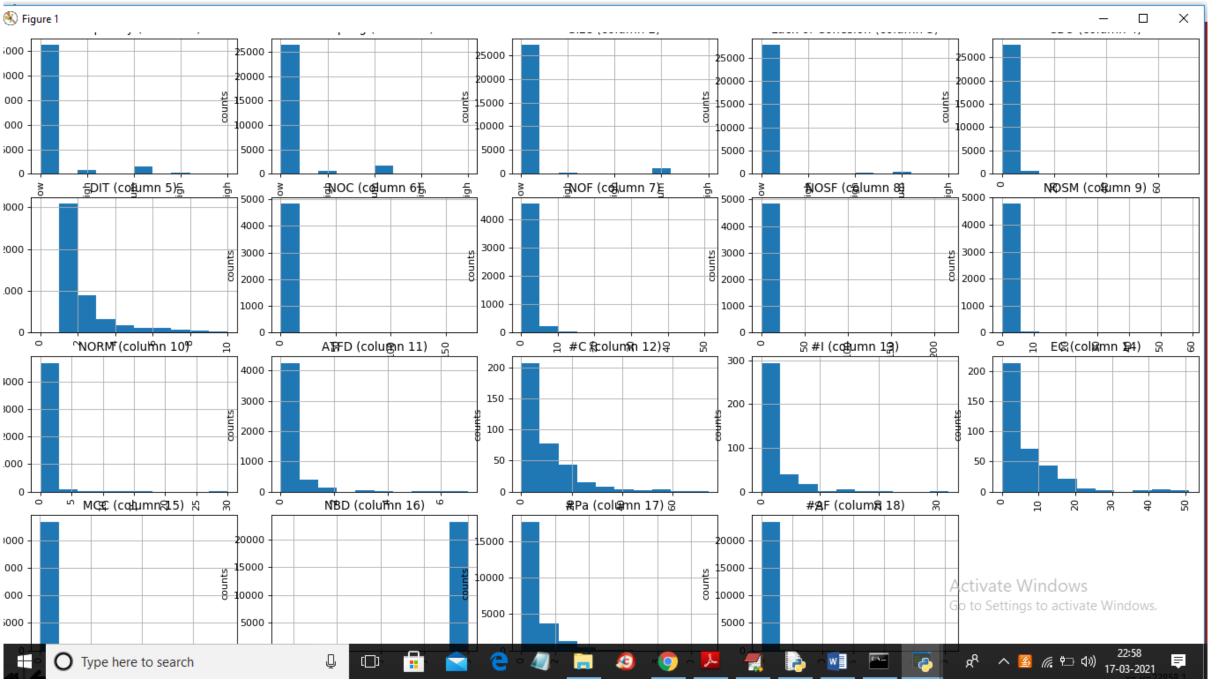
In above screen click on ‘Upload Dataset’ button to and upload dataset


An Experimental Study for Software Quality Prediction with Machine Learning Methods
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:)
Open
Cancel
Activate Windows
Go to Settings to activate Windows.
Type here to search
22:56
17-03-2021

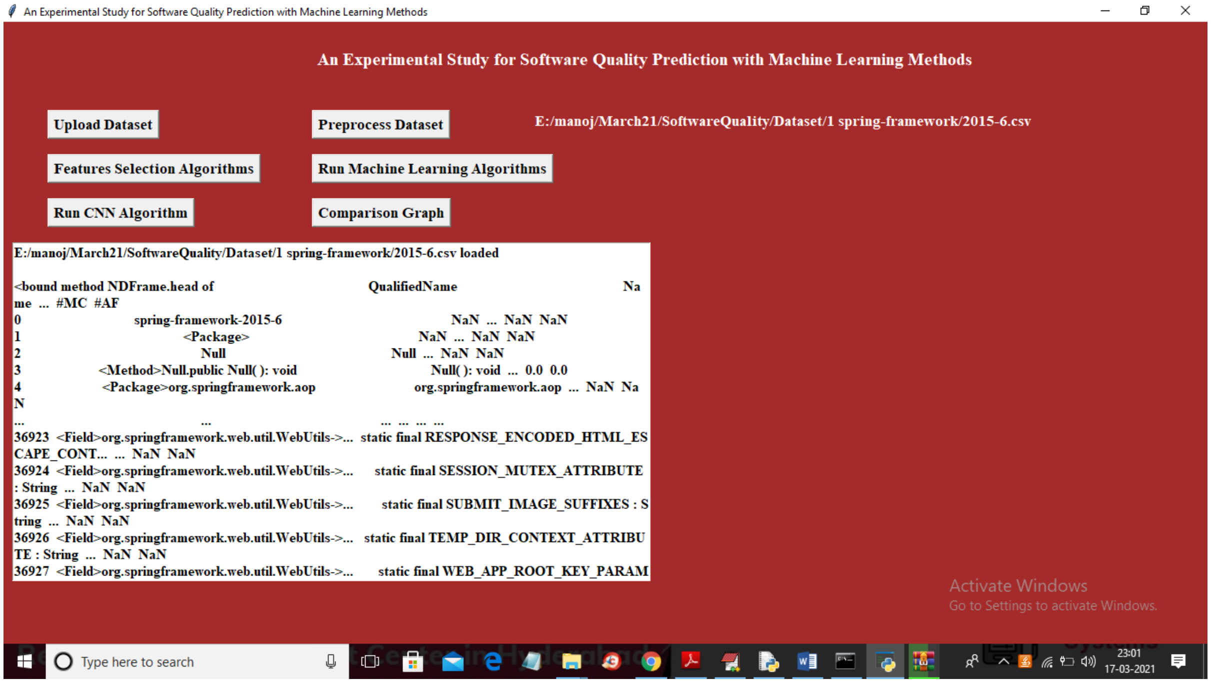
Name	Date modified	Type
2015-6.csv	15-07-2020 04:09	Microsoft Excel C
2016-6.csv	15-07-2020 04:09	Microsoft Excel C
2017-6.csv	15-07-2020 04:09	Microsoft Excel C
2018-6.csv	15-07-2020 04:09	Microsoft Excel C
2019-6.csv	15-07-2020 04:09	Microsoft Excel C
2020-6.csv	15-07-2020 04:09	Microsoft Excel C
2020-7.csv	15-07-2020 04:09	Microsoft Excel C

File name: 2015-6.csv

In above screen selecting and uploading ‘2015-6.csv’ dataset file and then click on ‘Open’ button to load dataset and to get below screen



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

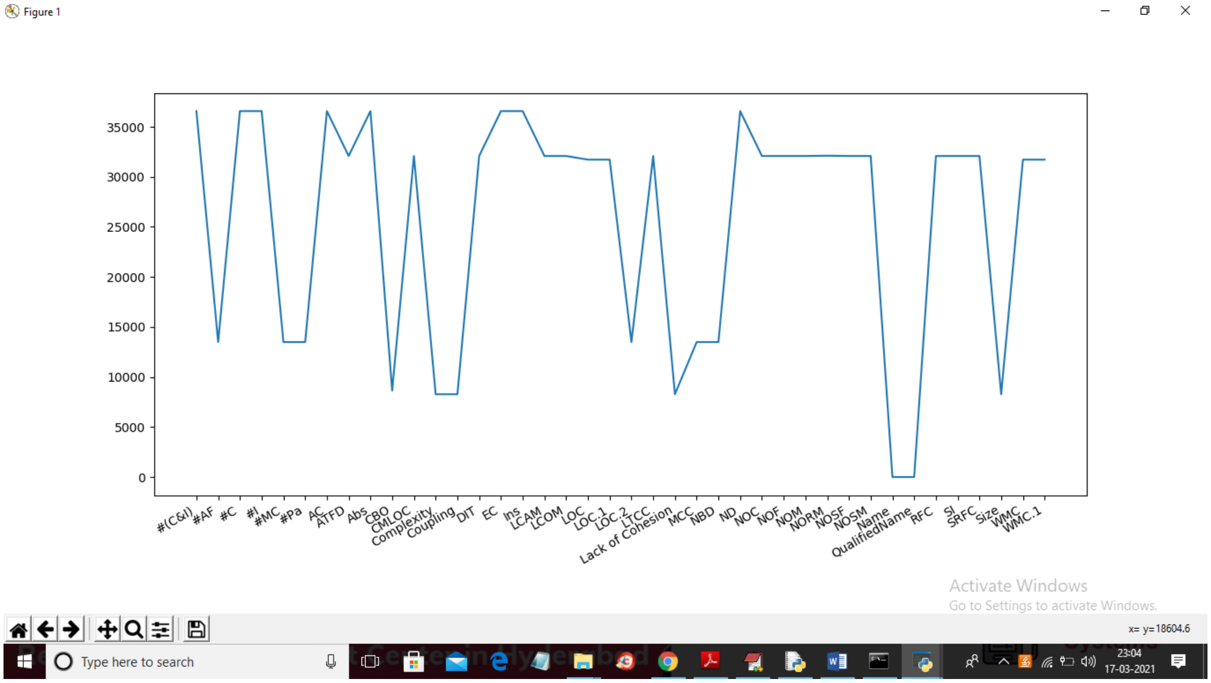

An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
23:01
17-03-2021

An Experimental Study for Software Quality Prediction with Machine Learning Methods

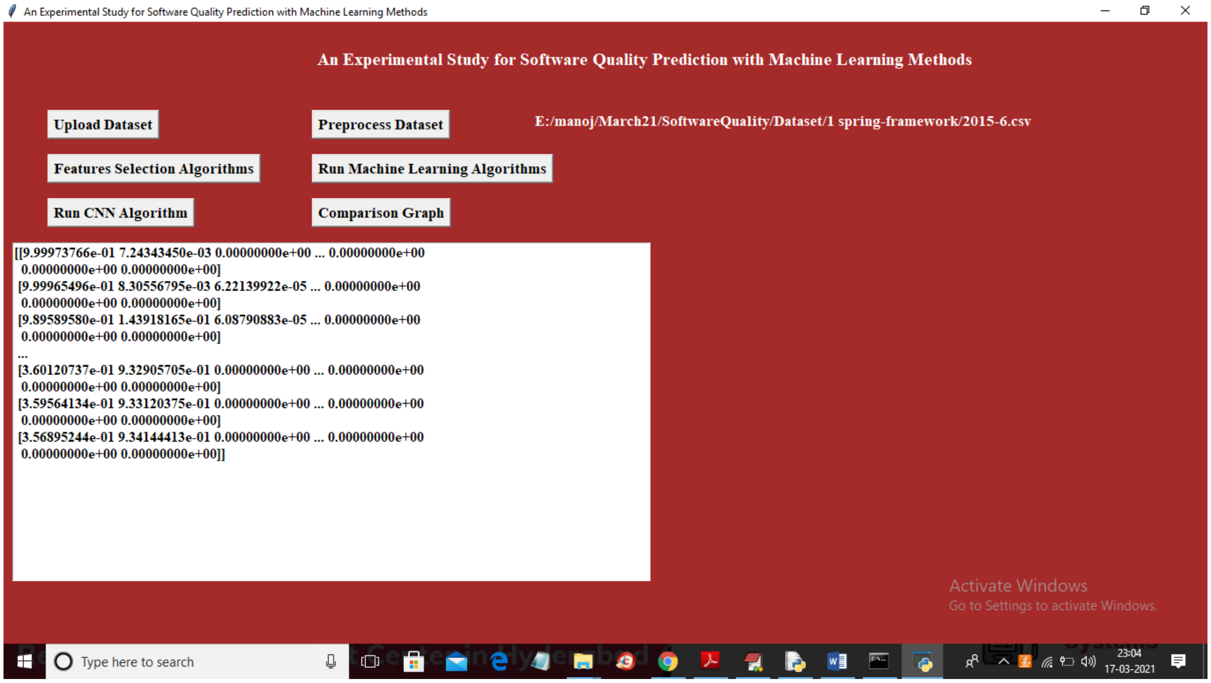
E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv

```
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
```

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



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

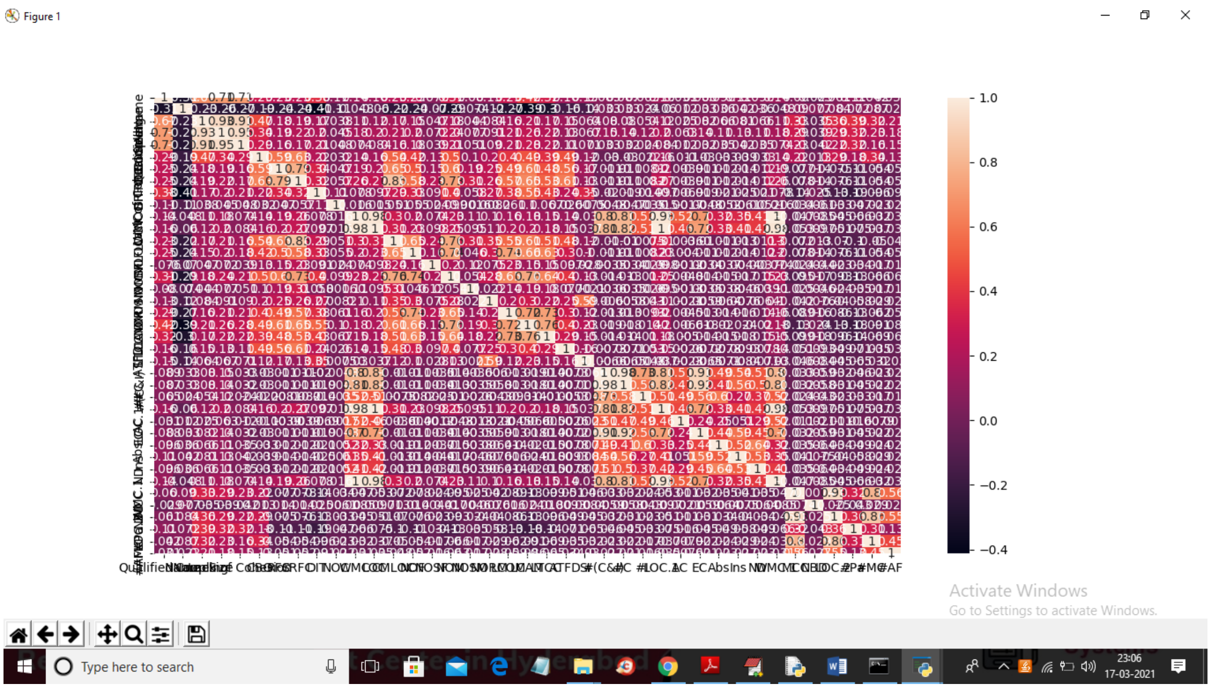

An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Type here to search
23:04
17-03-2021

An Experimental Study for Software Quality Prediction with Machine Learning Methods

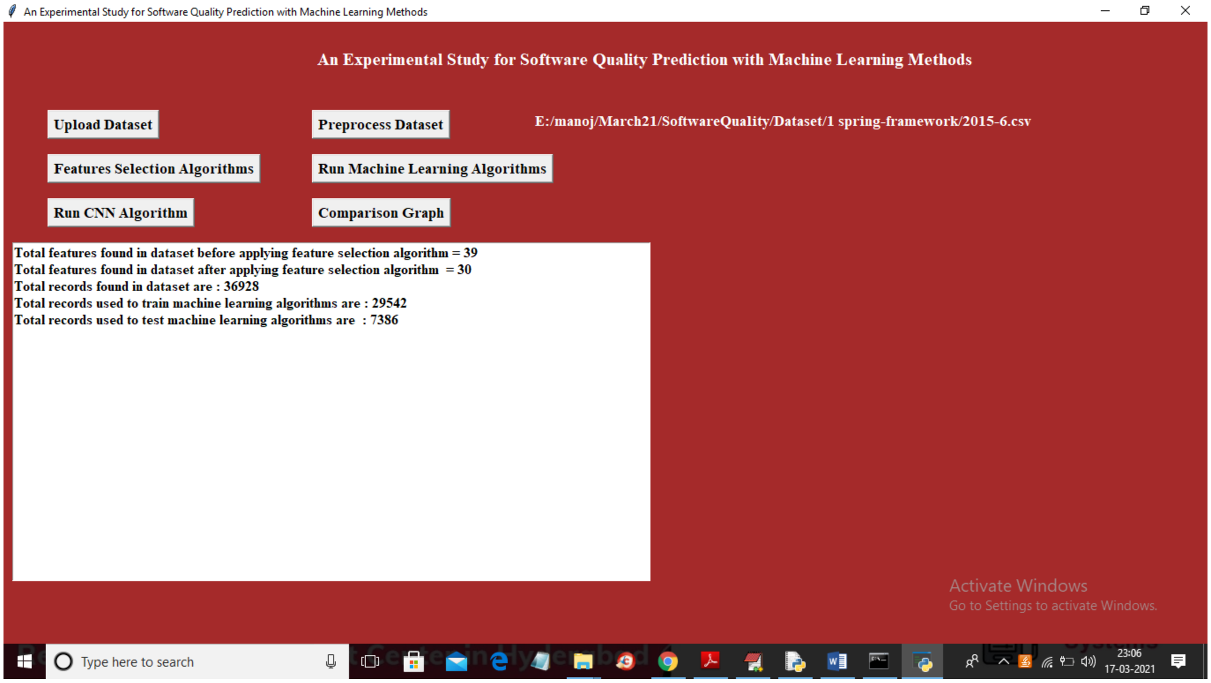
E:/manoj/March21/SoftwareQuality/Dataset/1 spring-framework/2015-6.csv

```
[[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]]
```

In above screen all missing an string values are replace 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



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

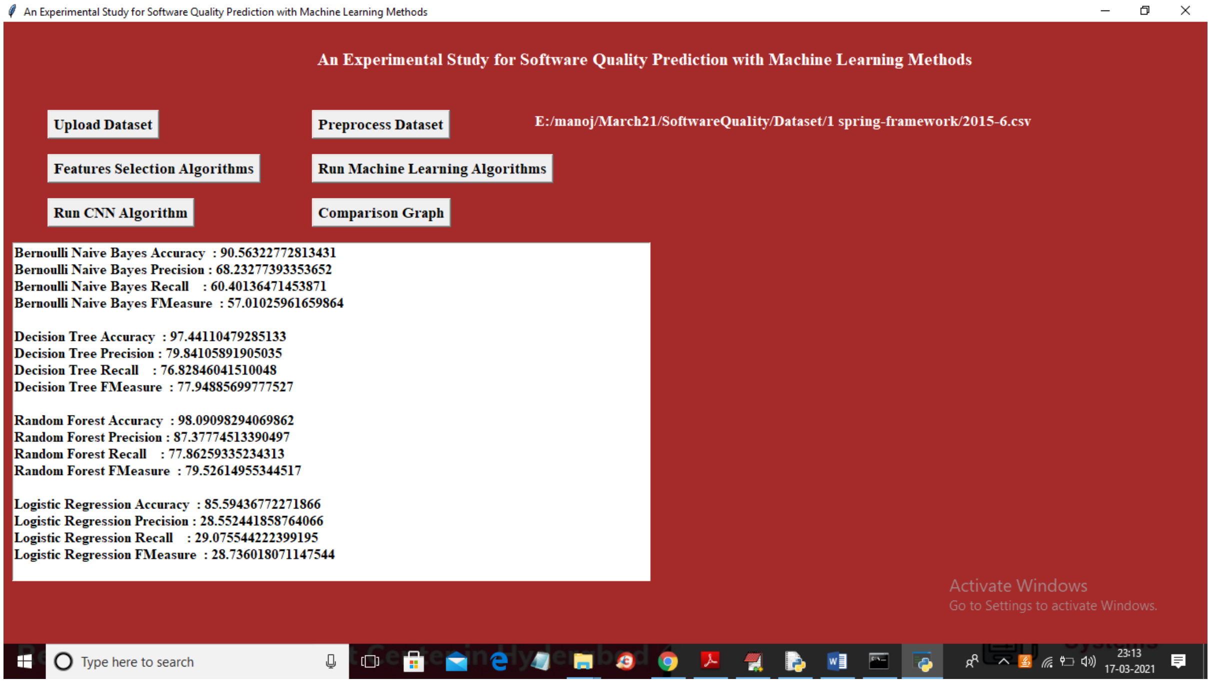

An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
23:06
17-03-2021

An Experimental Study for Software Quality Prediction with Machine Learning Methods

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

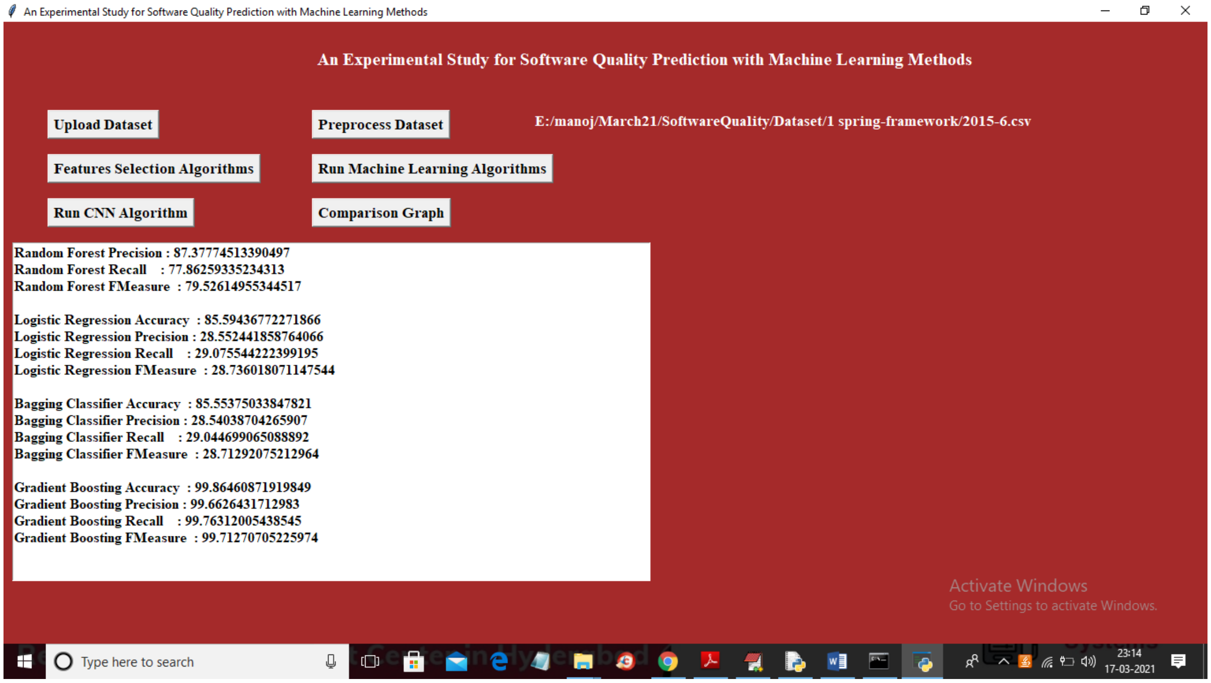
```
Total features found in dataset before applying feature selection algorithm = 39
Total features found in dataset after applying feature selection algorithm  = 30
Total records found in dataset are : 36928
Total records used to train machine learning algorithms are : 29542
Total records used to test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

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
Activate Windows
Go to Settings to activate Windows.
Type here to search
23:13
17-03-2021

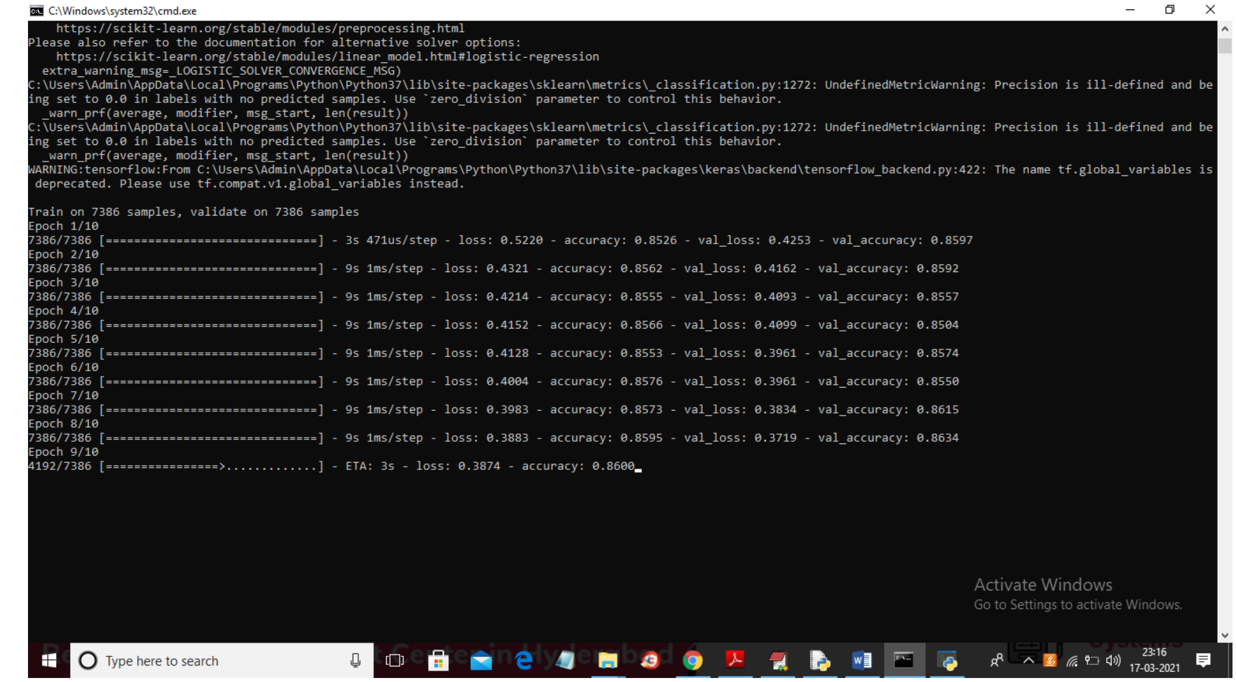
```
Bernoulli Naive Bayes Accuracy  : 90.56322772813431
Bernoulli Naive Bayes Precision : 68.23277393353652
Bernoulli Naive Bayes Recall    : 60.40136471453871
Bernoulli Naive Bayes FMeasure  : 57.01025961659864

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
```

In above screen we can precision, recall, accuracy and fscore 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
Activate Windows
Go to Settings to activate Windows.
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
```

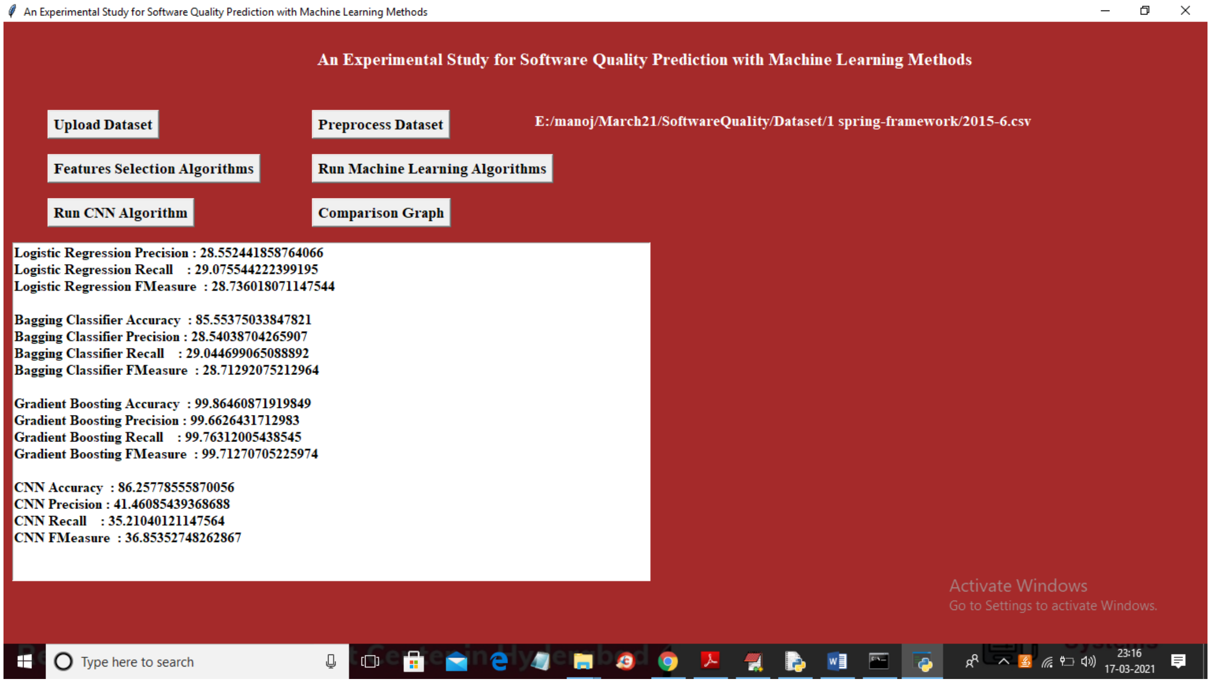
Now click on ‘Run CNN Algorithm’ button to run CNN algorithm and to get below screen



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

```
    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
```

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

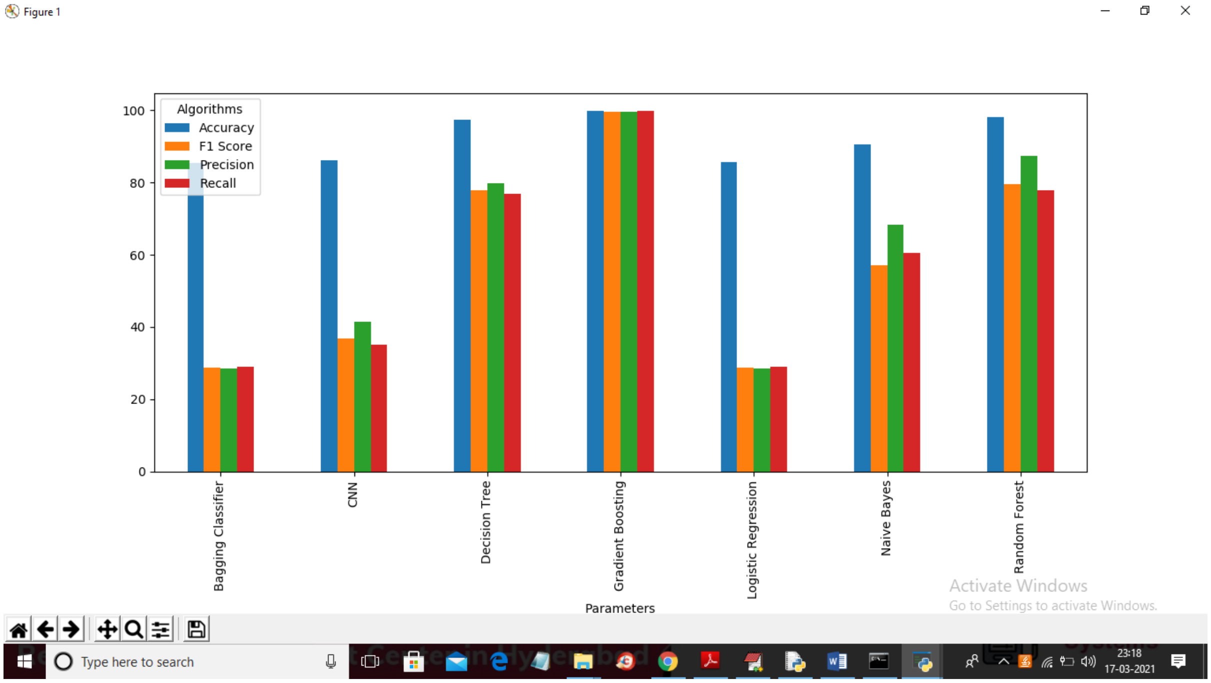


An Experimental Study for Software Quality Prediction with Machine Learning Methods
An Experimental Study for Software Quality Prediction with Machine Learning Methods
Upload Dataset
Preprocess Dataset
Features Selection Algorithms
Run Machine Learning Algorithms
Run CNN Algorithm
Comparison Graph
Activate Windows
Go to Settings to activate Windows.
Type here to search
23:16
17-03-2021

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

```
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
```

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



In above graph we are plotting accuracy, precision, recall and accuracy for each algorithm

10. CONCLUSION AND FUTURE ENHANCEMENT

In this paper we have experimented classification algorithms using Scikit-learn library on two dataset. We have experimented with recent algorithms that support multi-class classification. The accuracies achieved by using these algorithms are 92.28% on EBSPM Dataset and 92.22% on ISBSG Dataset. In comparison to previous directly comparable studies, acceptable level multiclass quality prediction could be achieved.

2. References

[1] Vijay, T. John, D. M. G. Chand, and D. H. Done. "Software quality metrics in quality assurance to study the impact of external factors related to time." International Journal of Advanced Research in Computer Science and Software Engineering, 2017.

[2] 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.

[3] X. Wang, Y. Zhang, L. Zhang and Y. Shi, "A Knowledge Discovery Case Study of Software Quality Prediction: ISBSG Database," 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Toronto, ON, 2010, pp. 219-222.

[4] X. Wang, Y. Zhang, L. Zhang and Y. Shi, "A Knowledge Discovery Case Study of Software Quality Prediction Based on Classification Models: ISBSG Database," The 11th International Symposium on Knowledge Systems Sciences (KSS 2010), 2010

[5] E. Rashid, S. Patnaik, and V. Bhattacherjee, "Software quality estimation using machine learning: Case-Based reasoning technique, " International Journal of Computer Applications, 2012

[6] www.isbsg.org

[7] https://goverdson.nl/

[8] H. Huijgens,"Evidence-based software portfolio management: a tool description and evaluation", 20th International Conference on Evaluation and Assessment in Software Engineering (EASE '16), 201