International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Development of a Simple Graphical Interface Based Software for Machine Learning and Data Visualization

Ogunleye G.O., Fashoto S.G, Daramola C.Y., Ogundele L.A., Ojewumi T.O., Adewole Timilehin

Abstract: ***Machine learning has become one of the foremost techniques used for extracting knowledge from large amounts of data. The programming expertise required to implement machine learning algorithms has led to the rise of software products that simplify the process. Many of these systems however, have sacrificed simplicity as they evolved and included more features. In this study, a machine learning software with a simple graphical user interface was developed with a special focus on enhancing usability. The system made use of basic graphical interface elements such as buttons and textboxes. Comparison of the system with other similar open-source tools revealed that the developed system showed an improvement in usability over the other tools.***

Keywords: GUI, Software, Machine Learning, data visualization

I. INTRODUCTION

Due to the increase in data generating devices and systems, there has been an exponential increase in the amount of data stored in electronic format in the past few decades[23]. As the volume of data being generated is increasing, so is the amount of effort required for human beings to manually process them. This led to the introduction of machine learning, an attempt to automate the data analysis and processing procedures.

Machine learning can be defined as the art of giving machines the ability to detect meaningful patterns in data and make decisions based on these patterns without being explicitly programmed [1]. This involves giving the machine a training data, generating a model from this training data and then using this model to predict future data. This is very useful for tasks that are too complex for explicit programming and tasks that require adaptivity.

The field of Machine learning is one of the fastest growing fields of computer science, with a wide range of applications These applications include, but are not limited to: speech recognition (Deng and Li[3], fraud detection [1], recommender systems [5], anti-spam filters, medical diagnosis[3], bioinformatics [2], and computer vision. There are many algorithms available to perform these tasks. Examples are classification algorithms such as K-nearest neighbors, Support Vector Classifier and Naïve Bayes classifier; Regression algorithms such as Linear Regression and Support Vector Regression;Clustering algorithms such as Mean Shift and KMeans; and deep learning algorithms such as the Artificial Neural Network. The technical know-how required to implement the various algorithms available in machine learning makes it a challenge for non-programmers to be able to make use of them. Therefore, it is necessary to provide a graphical user interface (GUI) based application which hides the implementation details of these algorithms from the user and also provide visualizations, therefore making it easy for non-technical users to use. There are various existing open source software packages that provide graphical user interface functionalities for machine learning tasks such as WEKA, RapidMiner, KNIME, and Orange.

The Waikato Environment for Knowledge Analysis, WEKA [25] is one such system. It provides easy access to machine learning tasks such as data preprocessing, classification, regression, clustering, visualization and feature selection with both GUI and command line interface.

RapidMiner Studio [4] is a similar platform developed by RapidMiner. It has both proprietary and open source versions. It also provides an integrated environment for easy preprocessing, machine learning, and predictive analytics. However, some of its features are limited in the open source version.

KNIME (Konstanz Information Miner) is a data analytics, reporting and integration platform that uses modular data pipelining to implement its machine learning components.

While these applications have their advantages, the steep learning curve associated with them due to their large amount of functionalities and also the large amount of system memory they occupy has created the need to produce a simpler software focused on accommodating beginners. The primary purpose of this paper is to develop an application which simplifies machine learning tasks by abstracting them into simple GUI operations. The software presented in this paper will provide users without the knowledge needed to write programs the opportunity to perform basic machine learning and visualization tasks by simplifying these tasks into GUI operations. It will also help experienced machine learning programmers who wish to quickly build machine learning models for small tasks to save time.

Revised Manuscript Received on 30 July 2019.

***** Correspondence Author

Ogunleye G.O*., Department of Computer Science, Federal University, Oye-Ekiti, Ekiti State, Nigeria.

Fashoto S.G, Department of Computer Science, University of Swaziland, Kwaluseni, Swaziland

Daramola C.Y., Department of Computer Science, Federal University, Oye-Ekiti, Ekiti State, Nigeria

Ogundele L.A., Department of Computer Science, College of Education, Ilesa, Osun State, Nigeria

Ojewumi T.O., Department of Computer Science, Redeemer's University, Ede, Osun State, Nigeria

Adewole Timilehin, Department of Computer Science, Federal University, Oye-Ekiti, Ekiti State, Nigeria

The algorithms to be considered in the proposed software are limited to: i. K-nearest neighbors, ii.Support vector classifier, iii. Linear regression, iv. Ridge regression, v. Lasso regression, vi. Support vector regression, vii. K-Means, viii. Mean shift, ix. Artificial neural network.

II. RELATED WORKS

2.1 Overview

As computer technology is advancing, more people are making use of computer systems for various tasks. This implies that more people without much knowledge of programming need to perform tasks that require expert programming knowledge. As a result, many efforts have been made to solve this problem by creating software tools that can significantly simplify these tasks [13].

The tools created for machine learning have varied in many aspects such as type (library, platform, framework), intended users (beginners, experts), interface (CLI, GUI, API), domain (business, computational biology, statistics, general), licensing (open-source, proprietary) and functionality among many others. In the rest of this chapter, some of these characteristics are discussed and some existing tools are reviewed.

2.2 Features of Machine Learning Tools

2.2.1 Software type

Programming languages are the lower level tools for machine learning. The choice of programming language used for a machine learning project may have a huge impact on the results. This can be due to factors such as ease of use, speed, scalability, extensibility, availability of libraries, data structures and other programming structures [8]. Some of the commonly used programming languages are Python, R, C++ and Java.

Libraries and packages are collections of facilities such as functions and methods that can be called by the user. Libraries usually provide only discrete facilities, that is, their facilities are specialized for specific tasks.

Platforms and toolkits provide more functionality than libraries, often enough to complete whole projects. They can contain their own integrated development environments. Their features are loosely coupled, enabling the user to tie them together in various ways [20].

2.2.2 User Interface

User interface design is one of the central issues on the usability of a software having a huge impact on the learnability of the software [21].

Machine learning tools with command line interfaces (CLI) feature text editors where users can enter command in the language supported by the tool. Such tools use visual elements sparingly and are therefore more suited to experienced users. They offer more flexibility to the user but they feature a steep learning curve. Work done on CLI is reproducible by simply saving the commands as scripts. Scikit-learn [10] is an example of CLI tools.

Machine learning tools with graphical user interfaces are usually easier for non-technical users to work with, because they tend to focus on graphical presentations such as visualizations. They also require lesser programming knowledge compared to their pure library counterparts. A major downside of this is that the interface imposes some restrictions on the actions a user can perform. An example of this type of tool is RapidMiner [4].

Machine learning tools with application programming interfaces provide interfaces that external programs can link to. This enables users to be able to incorporate machine learning into their own programs. An example of this type of tool is the LIBSVM library written in C++ and C programming languages.

2.2.3 Domain of application

While some machine learning tools are made for general purpose use, others are made for domain specific tasks. There is also a subset that are made for specific domains but can be used for general purpose tasks. Orange (Demsar et al.[14]) and Scikit-learn [10] are examples of general purpose tools while KNIME [17] is a tool with focus on computations in Chemistry but with general purpose applications.

2.2.4 Target users

Tools made for non-technical users tend to be more intuitive, visually-oriented and less technical. Such tools also feature reduced functionalities in order to avoid feature clutter. This ensures that simplicity, ease of use and learnability are maintained.

Tools made for experienced users on the other hand usually favor functionality and flexibility more than ease of use and learnability.

2.3 Existing Machine Learning Tools

In the past couple of decades, machine learning has become a popular method for performing tasks that require the extraction of information from datasets [6]. Several software tools have been developed to make machine learning programming easier and faster. They are in the form of:

i) Programming languages,
ii) Libraries/Packages,
iii) Graphical User Interface-based applications, platforms and toolkits,
iv) Combinations of any of these.

2.3.1 Programming Languages

Python

Python is one of the most popular programming language for machine learning because of its simple syntax, its modular architecture, and its huge repository of libraries. It can be challenging though for beginners to master the language because of its steep learning curve.

R Statistical Programming Language

R is a similar language designed mainly for statistical computing, data mining and machine learning. This domain-specific nature of the language together with its huge repository of libraries positioned it among the most widely used languages for machine learning [9]. Like Python, it can also be difficult to use for beginners.

2.3.2 Machine Learning Libraries

Scikit-learn

Pedregosa et al. [10] developed Scikit-learn, a python module which contains an array of advanced machine learning algorithms for medium-scale supervised and unsupervised problems. It contains several classification, regression and clustering algorithms including support vector machines and random forests, and is designed to work with NumPy and SciPy, the Python numerical and scientific libraries.

The main focus of the Scikit-learn package is to make machine learning more accessible to non-specialists using a general-purpose high-level language. Although, to use the package, a good knowledge of python programming is required, it still manages to considerably simplify the process by abstracting complex multi-line code into simple single-line method calls.

JSAT

Edward Raff [11] presented Java Statistical Analysis Tool (JSAT), a machine learning library written in Java and available under the GNU General Public License (GPL). It implements more than a hundred algorithms for classification, regression, clustering and visualization using an object-oriented framework. The main motivation for developing JSAT was to create a tool in the Java environment which is relatively fast and easy for developers, and provides further flexibility for researchers to create and compare new algorithms. Like its counterpart in Python – Scikit-learn – it also requires some programming knowledge to use.

MLPY

Mlpy is an open source machine learning library built on the Python NumPy, Scipy and GNU scientific libraries. It provides a variety of state-of-the-art machine learning techniques to solve both supervised and unsupervised problems. It is aimed at finding a middle ground between modularity, usability and efficiency. Although mlpy can be used for general purpose applications, its main focus is on computational biology [18].

MLR

Mlr is a package which provides a generic, object-oriented, and extensible framework for machine learning in the R programming language. It contains more than 160 modelling techniques for classification, regression, clustering and survival analysis. It also allows hyperparameter tuning, feature selection and visualizations [19]. The main targets of the mlr package are experienced practitioners and researchers.

2.3.3 **Graphical User Interface (GUI) Tools**

Waikato Environment for Knowledge Analysis (WEKA)

WEKA is a machine learning workbench developed in Java [25]. It provides an integrated environment with an interactive interface, a variety of machine learning algorithms, and various preprocessing, experimentation, and post-processing tools. It is one of the most popular machine learning tools for researchers and end users because of its exhaustive collection of algorithms, its ease of use due to its GUI, and its availability for free under the GNU GPL license [12]. In addition to WEKA's GUI, it also provides a simple Command Line Interface (CLI) which allows direct execution and also gives an expert user more freedom than the GUI.

Orange

Orange is a component-based toolbox for machine learning, data mining and visualizations written in Python under the GPL license [14]. Orange uses Python's open source libraries like Numpy, Scipy and Scikit-learn for its basic functionalities. It also uses the cross-platform Qt library to implement its visual interface. Orange's architecture is made of a canvas interface which the user can create their data analysis workflow on by placing widgets on it. The widgets provide the basic functionalities of the software and they are grouped into 'data', 'visualize', 'classify', 'regression', 'evaluate' and 'unsupervised'. In addition to the visual interface, Orange also contains a scripting interface which allows experienced programmers to use the software as a library.

RapidMiner

RapidMiner is a Java-based data mining tool with an IDE for machine learning, data mining, text mining, predictive analysis and business analytics [4]. It provides a visually appealing GUI which allows the user to simply connect visual components called operators together to form processes. RapidMiner supports about twenty-two file formats [15]; most types of databases; more than a hundred learning schemes and more than 1,500 techniques for data integration, transformation, analysis, modelling, visualization, attribute selection, outlier detection and parameter optimization.

KNIME (Konstanz Information Miner)

KNIME is an open source software platform for data analytics, reporting and integration [17]. It is based on the Eclipse platform. It enables the user to create workflows by connecting visual components, execute the workflows and analyze the results. KNIME was made with focus on Pharmaceutical research such as molecular analysis therefore it contains packages specifically made for such purposes.

Rattle

Rattle is a free open source GUI package for statistical analysis and model generation using the R statistical programming language. It is very suitable for learning machine learning with the R language because of the simple user interface it provides and because of its Log Code tab which generates the R script for all user's interactions with the GUI components, which can be copied by the user. It also provides a facility for partitioning the dataset into training, validation and testing sets [16].

2.4 **Comparison of Tools**

Some studies have been carried out to compare existing machine learning tools using characteristics such as functionality, usability and extensibility.

Chen, Williams, and Xu [22] conducted a survey and evaluation of twelve commonly used open source data mining and machine learning tools. ADAM, AlphaMiner, Databionic ESOM, Gnome Data Miner, KNIME, Mining Mart, MLC++, Orange, Rattle, TANAGRA, Weka and YALE (RapidMiner) were chosen for the survey. Features such as general characteristics, data source accessibility, data mining functionality, extensibility, and usability of each software were used as the criteria for evaluation. The study showed that each software had its own strengths and weaknesses. YALE was found to support the most number of data sources, AlphaMiner, KNIME and Weka were found to contain the highest amount of functionality and better usability.

Rangra and Bansal [23] conducted a comparative study of data mining tools. They presented a comprehensive and theoretical analysis of six tools including RapidMiner, Orange, KNIME, Weka, KEEL, and R. After the description of the specifications, features, advantages and disadvantages of the tools, it was concluded that KNIME and Weka are most suitable for non-technical users, while RapidMiner and Orange are more suitable for advanced users.

Saravanapriya [24] conducted a survey of open source data mining tools including Orange, RapidMiner, Weka, JHepwork and KNIME. The features and uses of each tool were discussed in the research.

III. METHODOLOGY

3.1 System Architecture

The system architecture presents the conceptual view of the proposed system showing the various components and their interconnectivity. Figure 1 below shows the architecture of the proposed system.

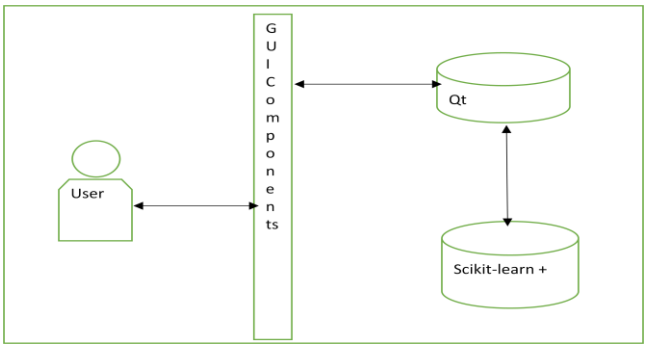

Figure 1 High Level System Architecture

3.2 Use Case Diagram

The use case diagram shows the different actors in the system and the actions each can perform with the system.

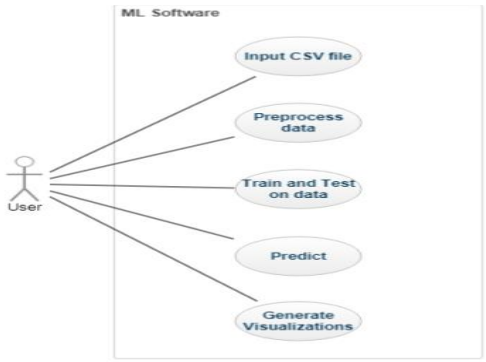

Figure 2 Use Case Diagram

3.3 Modular Structure

The system will be divided into five modules:

i. Preprocess: This module will handle the operations concerned with the loading of datasets, cleaning, selection and transformation of data. The module will also require the dataset to be loaded before other modules can be accessed.
ii. Classify: This module will contain the various classification algorithms.
iii. Regression: This module will contain the various regression algorithms.
iv. Cluster: This module will contain the various clustering algorithms.
v. Error handler: This module will handle the errors that occur during the operation of the system.

3.4 Algorithm

Step 0: Start

Step 1: Read dataset

Step 2: Preprocess dataset

Step 3: Select algorithm

Step 4: Build model

Step 5: if task category != clustering

a. Read features for prediction

b. Generate prediction

Step 6: Generate visualization

Step 7: Save model

Step 8: Stop

3.5 Flowchart

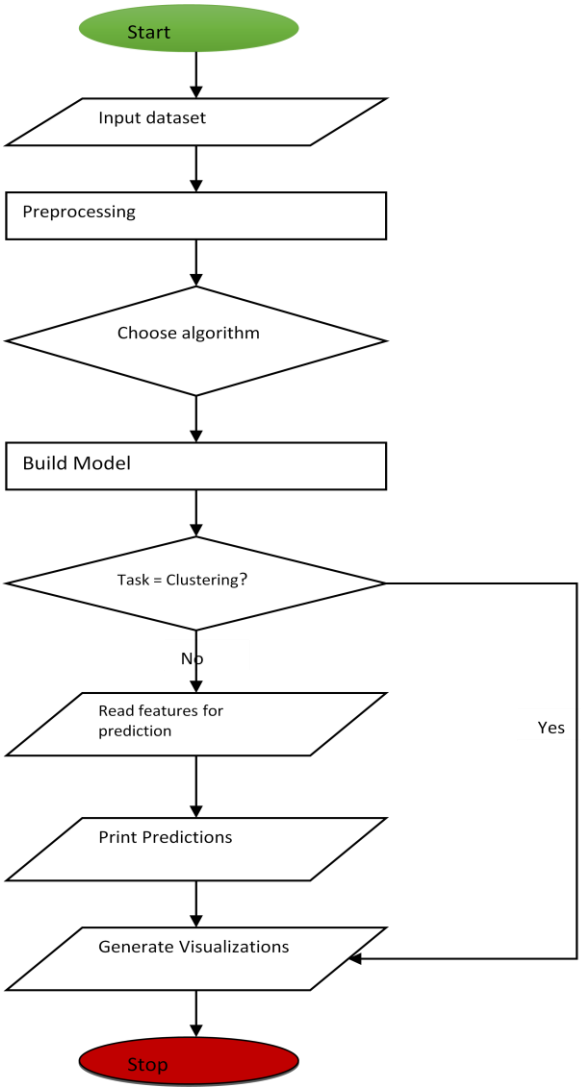

Figure 3: Flowchart for the Proposed System

IV. PLEMENTATION, RESULT AND DISCUSSION

4.1
Implementation Tools

4.1.1 **Programming Language and Libraries**

The programming language of choice for the implementation of the software is Python. Python is a modern language that is easy to write and easy to read because of its formatting style. It has a rich collection of libraries for machine learning and data visualizations.

4.2 **System Graphical User Interfaces**

Figure 4 shows the preprocessing tab where the user can load the data file, view some details about the data and apply preprocessing operations on the data.

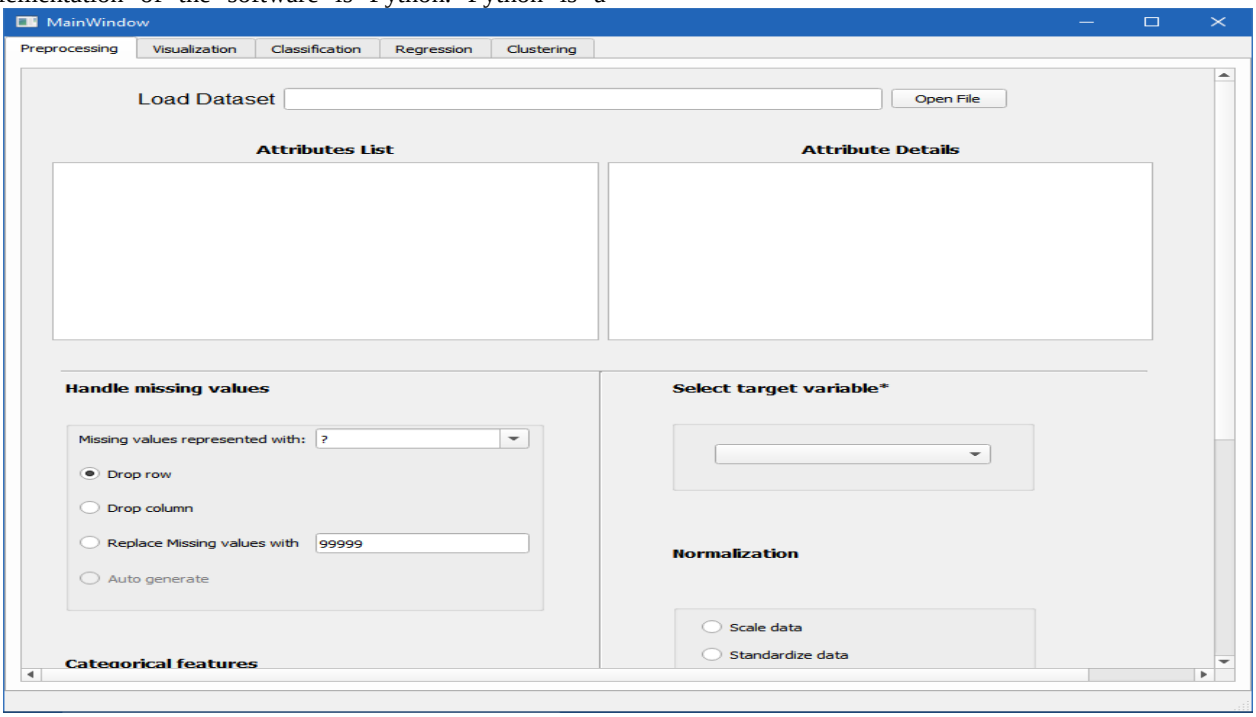

Figure 4: Preprocessing Tab

The data explorer shown in *Figure 5* enables the user to see the data in a tabular format. The data explorer is populated immediately the data is loaded.

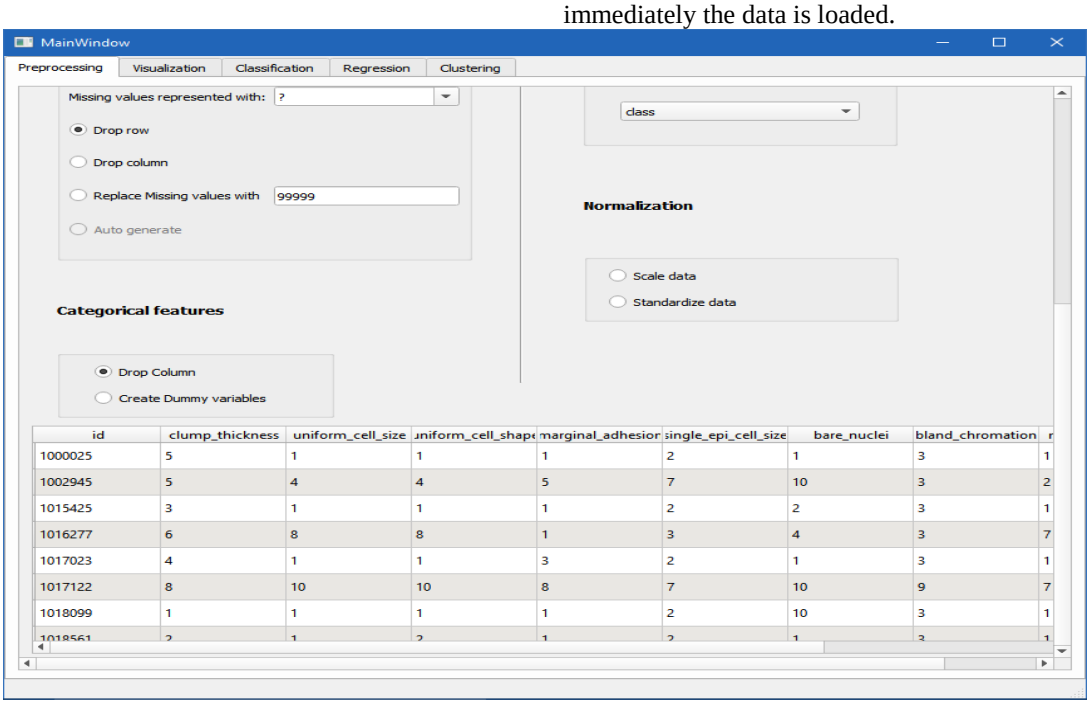

Figure 5: Data explorer

The visualization tab as shown in *Figure 6* enables the user to create scatter plots, line plots, and histograms. The user is able to select which features to plot from the input data.

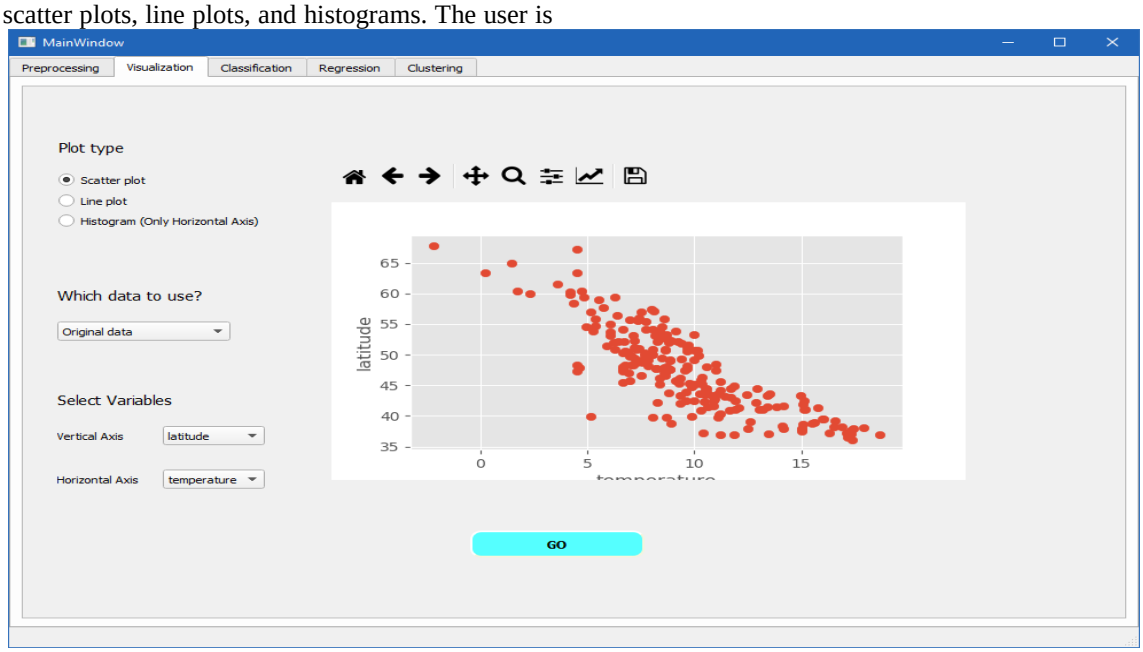

Figure 6: Visualization Tab

The classifications tab illustrated in *Figure 7* is where the user can select the algorithm parameters and perform model training, testing and prediction. A data explorer is also provided where the user can see the results of the prediction in tabular format.

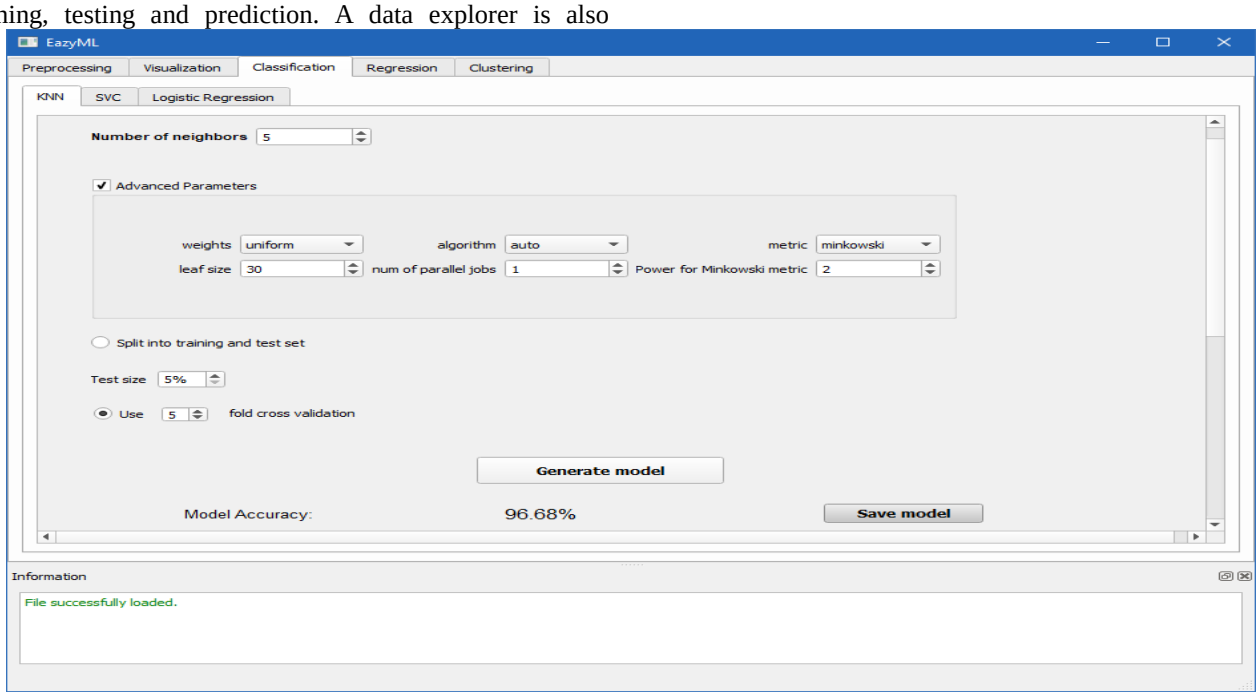

Figure 7: Classification Tab

4.1 Performance Comparison with Existing Systems

The testing process was divided into two phases. The first phase is the quantitative testing where metrics such as accuracy and runtime of the system were compared with those generated by Weka, Orange, RapidMiner, and KNIME on the same data. The results are shown in Table 1.

Table 1 Result of comparison on accuracy

	Developed Software	Orange	KNIME	RapidMiner	Weka
KNN	92.5%	86.1%	85.91%	98.59%	81.13%
SVC	100.0%	85.2%	85.92%	85.92%	Not Implemented
Logistic Regression	88.89%	84.3%	85.92%	87.79%	81.13%

The second phase of the testing is the qualitative testing where the developed system, Weka, Orange, RapidMiner, and KNIME were evaluated for its usability by a group of users consisting of both experienced programmers and non-programmers. The users were asked to evaluate the systems by filling the System Usability Scale.

Table 2 System Usability Scale scores

Developed Software	Weka	RapidMiner	Orange	KNIME
77.5	75.2	69.1	72.8	68.9

V. CONCLUSION

From the analysis and assessment of the developed software, it can be concluded that the simplicity approach taken its development has been able to improve its usability for novice users, thereby
giving the users more opportunities to take advantage of machine learning.

5.2 Recommendations

Future work should focus on providing more algorithms and advanced features while still retaining simplicity. Features that can be added include:

i. Pipelining
ii. Hyperparameter tuning
iii. Ability to edit data directly on the application
iv. More visualization types and more customization options
v. Metrics such as confusion matrix, specificity, sensitivity and Area under ROC curve
vi. Interactive tutor for beginners
vii. Artificial Neural Networks should be included

Similar software should also be developed for mobile operating systems such as Android and iOS.

REFERENCES

1. Richard Bauder and Taghi Khoshgoftaar. (2017).Medicare Fraud Detection Using Machine Learning Methods. 858-865. 10.1109/ICMLA.2017.00-48.
2. Baldi, P and Brunak, S, *Bioinformatics: The Machine Learning Approach*, 2nd edition. MIT Press, 2001. ISBN 0-262-02506-X.
3. Deng, L.; Li, Xiao (2013). "Machine Learning Paradigms for Speech Recognition: An Overview". *IEEE Transactions on Audio, Speech, and Language Processing*.
4. Gregor Guncar, Matjaz Kukar, Mateja Notar, Miran Brvar, Peter Cernelc, Manca Notar, Marko Notar (2018). "An application of machine learning to hematological diagnosis". Scientific reports 8(1), 411, 2018.
5. Markus Hofmann, Ralf Klinkenberg (2013), "RapidMiner: Data Mining Use Cases and Business Analytics Applications (Chapman and Hall/CRC Data Mining and Knowledge Discovery Series)," CRC Press, October 25, 2013.
6. Prem Melville and Vikas Sindhwani (2010). "Recommender Systems". Encyclopedia of Machine Learning, 2010.
7. Shai Shalev-Shwartz and Shai Ben-David (2014). "Understanding Machine Learning: From Theory to Algorithms". Cambridge University Press.
8. Sara Landset; Taghi Khoshgoftaar; Aaron Richter, Tawfiq Hasanin (2015). "A survey of open source tools for machine learning with big data in the Hadoop ecosystem". Journal of Big Data. Vol. 2.
9. Sylvia Tippmann (2014). "Programming tools: Adventures with R". Nature 517: 109-110. doi: 10.1038/517109a.
10. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Perrot, Édouard Duchesnay (2011). "Scikit-learn: Machine Learning in Python". Journal of Machine Learning Research. 12: 2825–2830.
11. Edward Raff (2017). "JSAT: Java Statistical Analysis Tool, a Library for Machine Learning". Journal of Machine Learning Research. 18: 1–5.
12. Eshwari Kulkarni and Raj Kulkarni (2016). "WEKA Powerful Tool in Data Mining". International Journal of Computer Applications 0975 – 8887.
13. A. Jović, K. Brkić and N. Bogunović (2014). "An overview of free software tools for general data mining".
14. Janez Demsar, Tomaz Curk, Ales Erjavec, Crt Gorup, Tomaz Hocevar, Mitar Milutinovic, Martin Mozina, Matija Polajnar, Marko Toplak, Anze Staric, Miha Stajdohar, Lan Umek, Lan Zagar, Jure Zbontar, Marinka Zitnik, Blaz Zupan (2013). "Orange: data mining toolbox in Python". JMLR. 14(1): 2349-2353.
15. Ralf Mikut and Markus Reischl(2011). "Data Mining and Knowledge Discovery". Wiley Interdisciplinary Reviews. Volume 1, Issue 5, pages 431–443, 2011.
16. Graham J Williams (2011). "Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery". Springer, Use R!.
17. Ian Witten and E. Frank (2005). "Data Mining: Practical machine Learning tools and techniques" 2nd edition. Morgan Kaufmann.
18. Davide Albanese, Roberto Visintainer, Stefano Merler, Samantha Riccadonna, Giuseppe Jurman, Cesare Furlanello (2012). "mlpy: Machine Learning Python". arXiv:1202.6548.
19. Bernd Bischl, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, Zachary M. Jones (2016). "mlr: Machine Learning in R". Journal of Machine Learning Research 17 1-5.
20. Jason Brownlee (2015). "Machine learning tools". Start Machine Learning Retrieved from https://machinelearningmastery.com/machine-learning-tools/
21. Alan Dix, Janet Finlay, Gregory Abowd, Russel Beale (2004). "Human-Computer Interaction". Third Edition. Pearson Prentice Hall.
22. Xiaojun Chen, Graham Williams, and Xiaofei Xu (2007). "A Survey of Open Source Data Mining Systems". Knowledge discovery and data mining, 2007. Springer.
23. Kalpana Rangra and K. Bansal (2014). "Comparative Study of Data Mining Tools". International Journal of Advanced Research in Computer Science and Software Engineering. Volume 4, Issue 6.
24. K. Saravanapriya (2014). "A Study on Free Open Source Data Mining Tools". International Journal of Engineering and Computer Science, Vol 3, Issue 12, Pages 9450-9452.
25. Geoffrey Holmes, Mark Hall, Eibe Frank, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009). "The WEKA Data Mining Software: An Update". SIGKDD Explorations, Volume 11, Issue 1.

AUTHORS PROFILE

FIRST AUTHOR: Dr. OGUNLEYE, Gabriel Opeyemi is a lecturer in the Department of Computer Science, Federal University, Oye-Ekiti, Ekiti State, Nigeria. He holds a Ph.D degree in Computer Science from The Federal University of Technology, Akure, Ondo State. His research interests include Machine learning, Computer Security, Mobile Agent Systems, and Artificial Intelligence among others.

SECOND AUTHOR: Dr. FASHOTO, Stephen Gbenga had his Ph.D from University of Ilorin, Nigeria in 2014. He is currently a lecturer at the Department of Computer Science, University of Swaziland, Kwalueni, Swaziland. He has published a number of articles in reputable journals and learned conferences.

THIRD AUTHOR:Comfort Y. Daramola (Mrs.) is a Lecturer in the Department of Computer Science. She obtained the Bachelor of Science (Education) degree from the University of Benin, Benin City and a Master of Science from the University of Ibadan. She is currently a Ph.D student in the Department of Computer Science, Federal University of Technology, Akure. Her research interest foci are Security, Bioinformatics and Information System

FOURTH AUTHOR: Dr. Ogundele Lukman Adebayo got his B.Tech Degree in Computer Science/Maths from Federal University of Technology, Minna. He had his masters and Ph.D degree in Computer Science from Obafemi Awolowo University and Federal University of Technology, Akure, Nigeria. His research activities spanned from Machine Learning, Artificial Intelligence and mobile agent systems.

FIFTH AUTHOR: Mrs. Ojewumi. Teresa O. is a programmer I in the Department of Computer Science, Redeemer's University, Ede, Osun State, Nigeria. She holds a masters degree in Computer Science from University of Ibadan, Oyo State. She is currently pursuing her Ph.D degree at Ladoke Akintola University of Technology, Ogbomoso. Her research interests include E-Learning, Fault Tolerant Systems among others.

SIXTH AUTHOR: Mr. Adewole Timilehin is a graduate of the Department of Computer Science, Federal University, Oye-Ekiti, Ekiti State, Nigeria.