Testcase dataset of an MIS system with priority

Ramya Jagarlamudi
Z1948509,Northern Illinois University

Rushith Kumar Challa
Z1948507,Northern Illinois University

Abstract: As in the world's present scenario the software development and software testing play's a vital role. By using software development applications we can create, design and deploy the software features. Which the software it contains set of instructions because of this development we can emerge a greater number of applications. Now for this Management Information system in the software testing we test the data type of a variables depends upon their priority for example like where the data testcase is high, low and then medium. Here we used some of the machine learning algorithms like random forest classifier, decision tree classifier and then logistic regression classifier. The testing fields present experts with the biggest obstacles because of this prioritizing of testcases for a particular dataset will be an advantage of time consuming and will be ease for solving of critical testcases in a software testing. It can be analyzed for future study will be motivated by current research's limits and outstanding problems. The study aims to examine the software testing has been applied.

Key words: Software testing, machine learning algorithms, software engineering, challenge.

Introduction

The Software Testing method includes running the software to see if all the components meet the customer's specifications. If you discover any issues, treat them as errors or defects. This phase is more expensive because, despite its importance and complexity in testing, only a small amount of time has been left. The data sets that are difficult to find are taken from a piece of code or historical data that cannot be analyzed by the general public. There is a lot of potential in the fields of software testing and data science, particularly machine learning. While leading examination on testcase prioritization particularly in starting phases of programming test cycle the way organizations put forth the boundaries in programming industry.

Software testers may prioritize their test cases to run the most important ones earlier in the regression testing process in order to reduce the cost of regression testing. A higher rate of fault detection in a test suite is one potential objective of such prioritization. Prioritization techniques can significantly increase the rate of fault detection, according to previous research. However, several additional inquiries were raised by those studies:1) Can techniques for

prioritizing specific modified versions be effective;2) What are the advantages and disadvantages of course versus fine granularity prioritization methods?

On the other hand, a lot of complicated software systems are built using a method called "component-based development," in which different parts come from different suppliers or teams within a company. Natural language is used to define test cases, and they are manually carried out. Regression testing becomes difficult in this situation because test case prioritization methods typically require access to source code. To the best of our knowledge, there are no methods that can use natural language test case descriptions to improve testing. To rank test cases, we use supervised machine learning (ML) and natural language processing (NLP). Testing's failure detection rate is improved as a result of earlier failure detection. This makes testing more effective than random and manual prioritization, especially since manually prioritizing test cases a priori is impossible and is done hardly.

We contribute the following in this paper:

- In the natural language processing the descriptions of black-box meta-data and test case is prioritizing by using a concept of regression testing. This improves the process of prioritizing test cases over random and manual ordering.
- In supervised machine learning to rank test cases for labelling of wanted and unwanted for training data of set cases is required which can be easily contributed.
- High dimensionality can result in significant reductions in prediction performance and potential long computation times due to the high number of features. In our instance, the size of the dictionary results in a significant increase in the number of features.

Prioritization of development activities considering stakeholder value propositions is central to value-based engineering. User perceptions of software quality are thought to rise when value-based software engineering practices are use with the goal of improving user perceptions of software quality, we investigate a value-driven strategy for prioritizing software system test. Software testing is a time-consuming and costly process.

We think adopting these four criteria to focus test efforts would help identify serious flaws more effectively and earlier in the software development process. We strive to find needs that would improve the client's perception of the quality of the program by concentrating on the priority that the customer has allocated.

A software project must dedicate at least 50% of its resources to testing [16]. Unfortunately, due to the restricted resources available for a typically extremely high number of testcases, it is too expensive to thoroughly test a software version. On the other hand, particularly safety-critical software systems need a thorough testing process, for example, to meet standards like ISO 26262.

Research Questions:

RQ1: Does R_Priority, weight, and complexity attributes play a role in determining the assigned testcases priority?

RQ2: Can we predict priority from these independent features using machine learning algorithms? Or are there any other ways for finding easy ways priority in software testing?

First, we need to find out the research questions that is there any relation for dependent and independent variables or is there any other way for finding the assigned testcases priorities. Secondly, we apply types of algorithms for finding priorities for different types of test cases in the software development and in software testing. Here in below we can see the block diagram of prioritizing of cases.

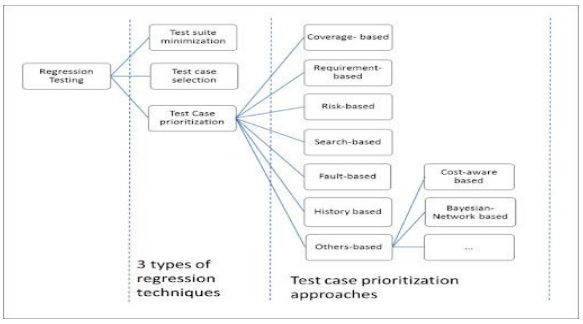


Fig 1: Block diagram of prioritizing of test cases

Limitations of study

One of the most difficult software testing tasks is known as "Test prioritization," and it enables testers to plan tests, consider cost values, manage risks, and analyze the tests that will be run in the context of a particular project. In our dataset based on the attributes for finding the each testcase of an MIS system based on the priority. Mainly in software testing for finding and detecting a bug there will be a greater number of test cases depends upon their priority, each testcase has its high, low, and medium priority values depends on requirement of product. The machine learning algorithms also solve the testcases based on the priority. There are some drawbacks of finding the priority of a data.

- Because of prioritizing of testcases it is difficult to handle complicated problems or issues.
- For this type of process it takes more time for an execution in a run-time application.
- It enables the testers to choose which tests to run to manage software delivery risks.
- We can say the other type of limitation is high cost expenditure.

Methodology

In this paper based on the research question we use some different types of methodologies they are

- Decision tree classifier algorithm
- Random forest classifier algorithm
- Logistic regression classifier

Decision tree classifier algorithm-A supervised machine learning algorithm called Decision Tree employs a set of rules to form opinions in a manner like how people do. An algorithm for machine learning classification can be thought of as being designed to make judgements. However the algorithm must choose which class to assign before the model can predict the class of a model, previously unseen input. The root and intermediary nodes are the independent features that are used for predicting useful information from the dataset and the leaf nodes are the target variable that needs to be predicted using the algorithm. This algorithm can be used in both the scenarios i.e., when the target feature has discrete values and when the

target variable has continuous values to be predicted

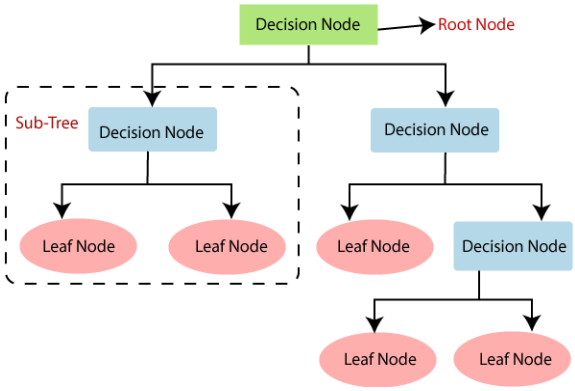
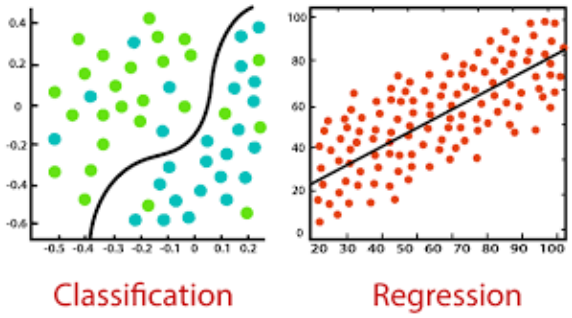


Fig 2: Block diagram of decision tree classifier

The decision tree classifier algorithm which it performs both the classification ad regression. In the machine learning we already understood about the classification and regression. Classification is used for finding the decisions to boundaries for splitting the dataset into different types of classes. Another algorithm is regression which is used for solving regression problems such as forecasting of various information.



An environment of supervised learning is used by Decision Tree. This means that the dependent variable (Y variable) in the data set used by the algorithm must be anticipated by knowing how it interacts with the numerous independent variables (X variables/predictors). In the block diagram we can observe that it consists of tree, sub-tress, leaf nodes and root nodes, the entire performance of this algorithm is in type of a tree. We need to know how this tree will work for that understand fully how decision tree's function, you must know ideas such as various node kinds, splitting, cleaning, attribute selection techniques, etc. But before things get complicated for a dataset.

Random Forest Classifier Algorithm

For a more precise prediction, Random Forest produces various decision trees that are then combined. The Random Forest model is based on the idea that several uncorrelated models (the various decision trees) work considerably better together than they do separately. The best results come from many very uncorrelated models (trees) working together as a group.

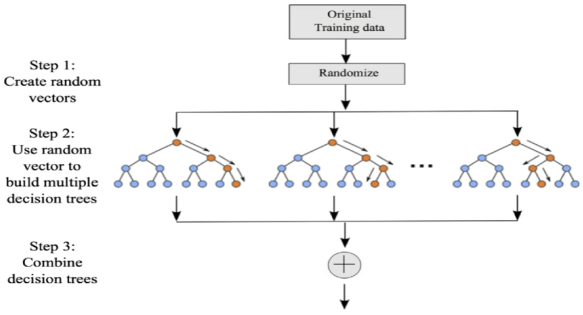


Fig 3: Block diagram of random forest classifier algorithm

We already known that the random forest classifier algorithm is like the decision tree classifier algorithm. This algorithm can combine multiple classifications to solve complex problems. In this type of algorithm first we train the original input data by creating random vectors in step 1 we can observe in the block diagram after the input data it randomize whole data and it splits the data into multiple decision trees based on the given training dataset. Combining the outputs of all decision trees we can find the

results. **What is the main purpose of using this type of algorithms in machine learning?**

By using this type of algorithm in machine learning gives us an effective output results and expected results. In these algorithms we train the data and test the data by finding the precision values, recall values and accuracy. By comparing all the algorithms accuracy values we can say that which algorithm will fit and predict the dataset.

Mainly the random forest classifier algorithm has three types of parameters. The parameters are total number of trees, total number of features and the third parameter is node size. In this algorithm the classification and regression problems are solved easily. The random forest algorithm is looks like a forest by combining all decision trees. Each tree in the algorithm in the is made up of a data sample called the bootstrap sample which can't come from a training dataset with replacement of other testing dataset. The out-of-bag (OOB) sample, which is one third of that training sample, serves as test data.

The prediction will be determined differently depending on the kind of problem. The individual decision trees for a regression task may be equals to the averaged, while for a classification of a particular task, a highly vote that is the predicted class will be the categorical variable with the highest frequency. Cross-validation using the OOB sample is then used to make that prediction final.

Logistic Regression classifier:

Logistic Regression is also a one type of classification technique used in machine learning. It uses a logistic function to model to find the variables. Among all types of algorithms this is the one base algorithm. Data can be described and in logistic regression there we can find relation between a single and one or more than one nominal, interval, ordinal, or we can find the ratio level of independent variables. In the binary regression model we have only one binary dependent variable coded by a variable used as an indicator and the other two values represents as 0 and 1. Whereas the independent variables will be having two different classes with an indicator same as dependent variable.

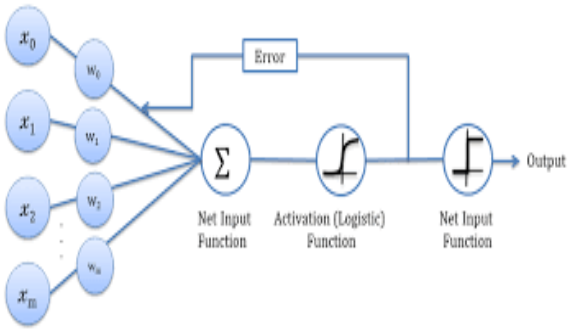


Fig 4: Schematic diagram of logistic regression classifier algorithm

In the above fig says that in logistic regression we have n number of input values we can assume the input like range, area, height, weight, standard deviation etc. There is a net input and net output functions in between them we have a logistic function after summing of all the inputs from the source we sum up them and pass the output of the net function as the input to the regression function in this function there we can observe the errors which again passes to

the input in a backward direction.

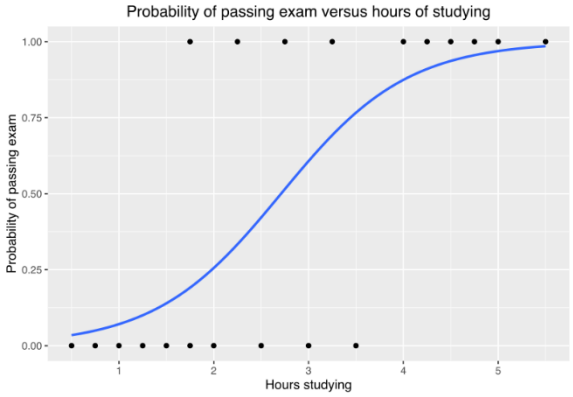


Fig 5: Sample results for logistic regression classifier

The fig says that sample results for logistic regression which is varying linearly at a particular point. It is an example how regression works.

Literature Review:

https://wwwiti.cs.uni-magdeburg.de › ps › auto

The authors in this paper explains about regression models of the software that is updated (changing or extending) then as the testing of the whole program is not practically possible, even though there is many test case prioritization that requires the source code. Generally, in the industrial field, system-level testing is used because it does not require any access to the source code. In this paper, the author used a better technique for test case prioritization based on supervised machine learning.

https://www.semanticscholar.org › paper

The authors Remo Lachmann, Manuel Nieke, Christoph Seidl, Ina Schaefer and Sandro Schulze are explaining about the prioritization of test cases in a system level by using various machine learning algorithms. The authors implemented a novel technology in this paper for testing in supervised machine learning. In this paper they find the errors easily by using techniques and by using these techniques there is a less chances of having failures in a test case.

https://taoxie.cs.illinois.edu › fse16industry-learning

The authors Benjamin Busjaeger and Tao Xie clearly explained about the there are some environments that were adopt as a large- scale continuos integration in this modern cloud-software that provides such as salesforce.com. so in this environments they majorly depend on the testing efficiently and effectively. So optimization under test prioritization is used for ranking tests and to run and identify the revealing failures. Here they present a novel approach for appyling the test prioritization in the industrial environments to integrate the multiple techniques of machine learning. So this kind of approach is very practical , well designed so efficient and it is well suited for the industrial settings.

https://dl.acm.org/doi/10.1145/566171.566187

The Author Jay Thiagarajan mentioned and explained about the development cycle the new defects in the software it should be detected as early as possible when the developers introduced. And they also built Echelon that works on the test prioritization and the approaches a binary code that varies on well to large systems. And here this Echelon processes very fast and provides well to compute the test that tests which will cover the basic blocks in the program. And then based on the results Echelon showed as it was well effective in test ordering based on versions.

Main Body of the paper

The main aim of our project is finding priorities of an MIS system based on their testcases in software testing. As we already known about how these priorities has been classified and it works for every testcase in testing.

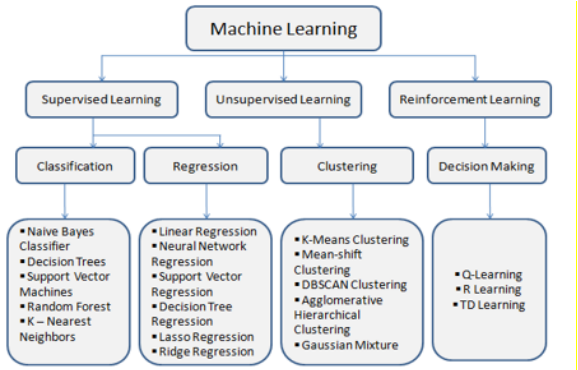


Fig: 2 Classification of Machine Learning algorithms

In the above figure says that the machine learning types and their classifications mainly we use two types of machine learning algorithms.

1. Supervised machine learning
2. Unsupervised machine learning

Mainly in supervised machine learning we have the classification and regression algorithms and in unsupervised learning we use only clustering algorithm. But in our project, we used only supervised machine learning algorithms. Our dataset consists of different types of attributes R_Priority, time, cost, weight, complexity, functional points etc. Depends on these we generate a research question is does R_Priority, weight, and complexity attributes play a role in determining the assigned testcases priority?

For that case we are finding the total number of rows and columns in our dataset by using various functions we are splitting out data into different slots after that we are checking whether there is an any imbalance of data is present in our dataset or not and then we are preprocessing our data which it means data cleaning.

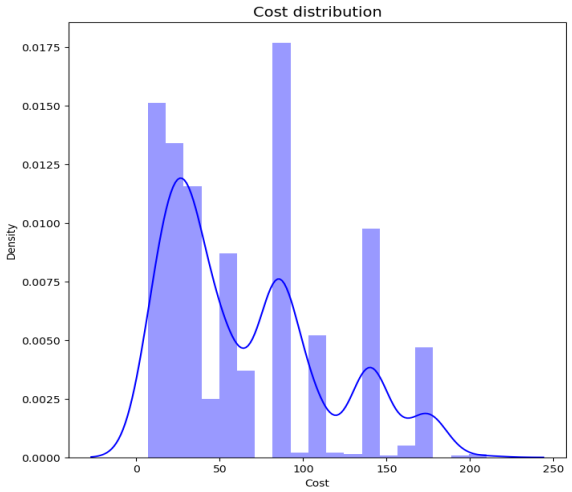


Fig 6: Distribution versus cost and density.

In the above figure it shows that the cost distribution between dependent and independent variables as cost and density. In the X-axis we taken cost as an independent variable and in y-axis the density as dependent variables.

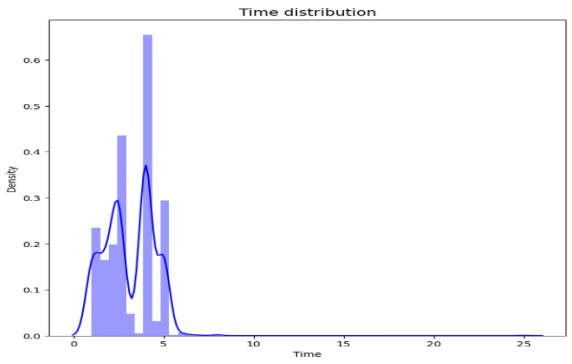
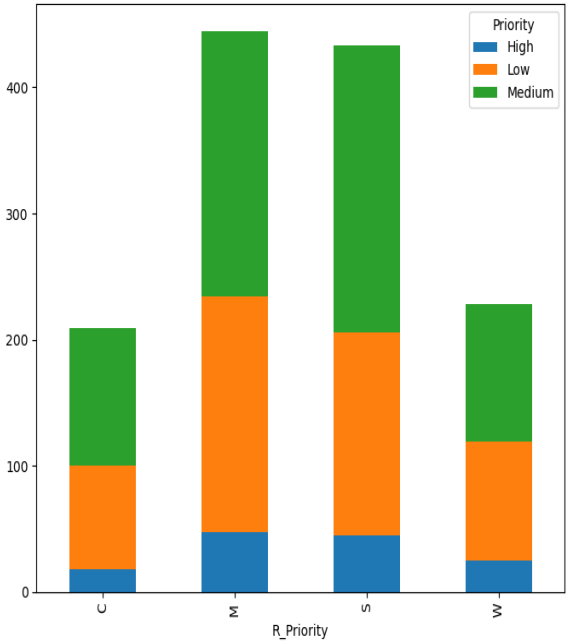
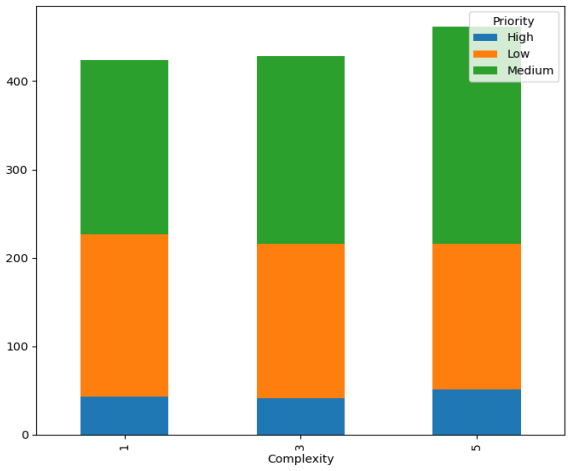
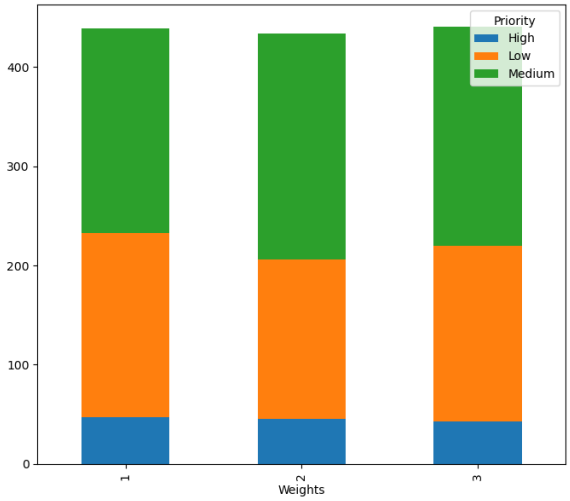


Fig 7: characteristics between density and time.

In the above graph we are plotting a graph that whether there is a any relation between the time and density. Depends on time the value of density is also increasing and decreasing for time interval. After certain time it moves constantly towards the axis.







For every requirement priority(Priority) there will be a high, medium, low priorities of an any type of business requirement models. If suppose we are taking independent variable as complexity for every interval of time it also contains the three priorities which is same as the weights but there is a difference in between them as sometimes there will be having more high priorities and less low priorities it is varying for different independent features of a system. For every R_Priority of values C,M,S,W and weights like 1,2,3.

In the below figure we are finding the correlation between the dependent and independent variables. As we can check that there is a correlation between R-Priority, weight, and complexity attributes plays a role in determining the assigned testcases priority in the dataset. As mentioned above we have used machine learning algorithms are decision tree classifier, random forest classifier and logistic regression.

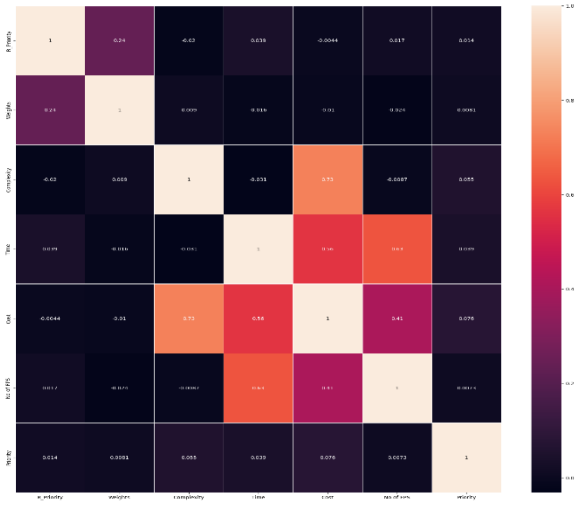


Fig 8: Heat map

In the below figure we are finding the correlation between the dependent and independent variables. As we can check that there is a correlation between R-Priority, weight, and complexity attributes plays a role in determining the assigned testcases priority in the dataset. As mentioned above we have used machine learning algorithms are decision tree classifier, random forest classifier and logistic regression.

We have used different algorithms functions and we predict and fit the data by finding precision, recall, accuracy and F1 score. For finding the precision we calculate the formula as

Precision (P) = True positive + False positive/ True positive.

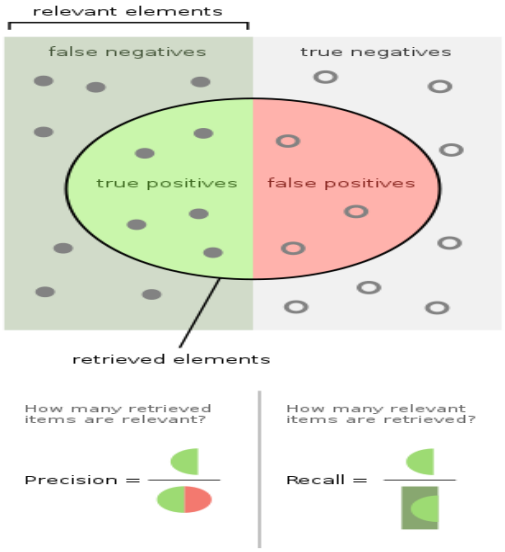
Recall (R) = True positive/ true positive + false negative.

Accuracy = TP+TN/ TP+TN+FP+FN

F1 Score:

F1 will be calculated by having precision and recall values

$$F_1 = \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$
$$= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$



In our dataset based on the attributes we find out the precision and recall values. For every different value of times there is a different weight, complexity, time values. Based on the positive and negative values we find the precision and recall values.

Results:

For Decision tree classifier:

	precision	recall	f1-score	support
0	0.5394	0.3746	0.4422	347
1	0.5117	0.6037	0.5539	434
2	0.5849	0.6263	0.6049	396
accuracy			0.5438	1177
macro avg	0.5453	0.5349	0.5337	1177
weighted avg	0.5445	0.5438	0.5381	1177

The about results shows the training data for decision tree classifier.

	precision	recall	f1-score	support
0	0.4167	0.3107	0.3560	177
1	0.5028	0.4072	0.4500	221
2	0.0732	0.2500	0.1132	36
accuracy			0.3548	434
macro avg	0.3309	0.3227	0.3064	434
weighted avg	0.4320	0.3548	0.3837	434

The above results say testing data for decision tree classifier.

For Random Forest classifier:

	precision	recall	f1-score	support
0	0.4393	0.2655	0.3310	177
1	0.5143	0.4887	0.5012	221
2	0.0855	0.2778	0.1307	36
accuracy			0.3802	434
macro avg	0.3463	0.3440	0.3210	434
weighted avg	0.4481	0.3802	0.4010	434

The above results show the random forest classifier outputs for precision, recall values and its accuracies of a training data.

	precision	recall	f1-score	support
0	0.5394	0.3746	0.4422	347
1	0.5117	0.6037	0.5539	434
2	0.5849	0.6263	0.6049	396
accuracy			0.5438	1177
macro avg	0.5453	0.5349	0.5337	1177
weighted avg	0.5445	0.5438	0.5381	1177

The above results show the random forest classifier for testing data**.**

For logistic regression classifier:

	precision	recall	f1-score	support
0	0.0000	0.0000	0.0000	347
1	0.3866	0.5968	0.4692	434
2	0.3866	0.4949	0.4341	396
accuracy			0.3866	1177
macro avg	0.2577	0.3639	0.3011	1177
weighted avg	0.2726	0.3866	0.3191	1177

The above results show the outputs for training data for logistic regression classifier with an accuracy**.**

	precision	recall	f1-score	support
0	0.0000	0.0000	0.0000	177
1	0.5000	0.5973	0.5443	221
2	0.1000	0.4722	0.1650	36
accuracy			0.3433	434
macro avg	0.2000	0.3565	0.2365	434
weighted avg	0.2629	0.3433	0.2909	434

The results show for logistic regression classifier for testing data.

Conclusion:

At the end of the section is conclusion we conclude that in software development and in software testing we are having many numbers of testcase In Management Information Systems we will have different number of test cases while doing testing. So, because of a greater number of tests cases, we can't find the priority of a particular case depends on high, low, medium. So, for easy way of understanding and finding the priority we took this type of dataset for better understanding. By using the machine learning we find out the accuracy of every priority of a business requirement and we check the heat map for the correlation between dependent and independent variables. We can predict Priority, weight, and complexity attributes play a role in determining the assigned testcases priority. And we did this analysis in SPSS, but we didn't get a prediction accuracy for the entire dataset. So, for the better future prediction we go through machine learning algorithms.

References:

[1] *System-Level Test Case Prioritization* Using Machine Learning. https://wwwiti.cs.uni-magdeburg.de/iti_db/publikationen/ps/auto/LachmannICMLA2016.pdf.

[2] Pan, Rong qi & Bagherzadeh, Mojtaba & Ghaleb, Taher & Briand, Lionel. (2022). Test Case Selection and Prioritization Using Machine Learning: A Systematic Literature Review. Empirical Software Engineering. 27. 10.1007/s10664-021-10066-6.

[3] Sharif, Aizaz, et al. "DeepOrder: Deep Learning for Test Case Prioritization in Continuous Integration Testing." ArXiv.org, 14Oct.2021,https://arxiv.org/abs/2110.07443

[4]] F. Wang, Y. Wang, X. Yang, Z. Hao and C. Zhang, "Machine Learning Transit Signal Priority Control of Bus Rapid Transit Based on Connected Vehicles Environment," *2021 6th International Conference on Transportation Information and Safety (ICTIS)*, 2021, pp. 152-156, doi: 10.1109/ICTIS54573.2021.9798410.

[5]] Konsaard, P.; Ramingwong, L. Using artificial bee colony for code coverage based test suite prioritization. In Proceedings of the 2015 2nd International Conference on Information Science and Security (ICISS), Seoul, Korea, 14–16 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–4.

[6] Khan, S.U.R.; Lee, S.P.; Parizi, R.M.; Elahi, M. A code coverage-based test suite reduction and prioritization framework. In Proceedings of the 2014 Fourth World Congress on Information and Communication Technologies (WICT), Malacca, Malaysia, 8–11 December 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 229–234.

[7] S. Elbaum, A. Malishevsky, and G. Rothermel. Incorporating varying test costs and fault severities into test case prioritization. Proc. Int'l Conf. Software Engineering, 2001.

[8]*Regression Testing Minimization, Selection and Prioritization: A Survey* https://onlinelibrary.wiley.com/doi/pdf/10.1002/stvr.430.

[9] C. A. Bhuvaneswari, M. Muthumari, A. J. S. Pragjnay, J. S. Lakshmi and T. Navya, "Design and Implementation of Remote Health Monitoring System and Application for Priority Recognition Using Machine Learning," *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, 2022, pp.1-5,doi: 10.1109/ICDSIS55133.2022.9915874.

[10] *The Application of Machine Learning in Test Case Prioritization - a Review.* https://www.researchgate.net/profile/Kleona-Binjaku/publication/338678675_The_Application_Of_Machine_Learning_In_Test_Case_Prioritization_-_A_Review/links/5eda4957299bf1c67d41cec2/The-Application-Of-Machine-Learning-In-Test-Case-Prioritization-A-Review.pdf.

[11] Andrei Paleyes University ofCambridge Department of Computer Science and Technology, et al. "Challenges in Deploying Machine Learning: A Survey of Case Studies." *ACM Computing Surveys*, https://dl.acm.org/doi/abs/10.1145/3533378.

[12] Marijan, Dusica. "Comparative Study of Machine Learning Test Case Prioritization for Continuous Integration

Testing." *ArXiv.org*, 22 Apr. 2022, https://arxiv.org/abs/2204.10899.

[13] *Learning for Test Prioritization: An Industrial Case Study*. https://taoxie.cs.illinois.edu/publications/fse16industry-learning.pdf.

[14] A. Srivastava and J. Thiagarajan. Effectively prioritizing tests in development environment. In Proc. ISSTA 2002, pages 97–106.

[15] Lachmann, R.; Schulze, S.; Nieke, M.; Seidl, C.; Schaefer, I. System-Level Test Case Prioritization Using Machine Learning. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 361–368.

[16] Jeyaprakash, S., Alagarsamy, K. (2015). A Distinctive Genetic Approach for Test-Suite Optimization. Elsevier International Conference on Soft Computing and Software Engineering, 62, 427-434.

[17] Gantait, A. (2011). Test case Generation and Prioritization from UML Models. 2 nd IEEE International Conference on Emerging Applications of Information Technology, 345-350.

[18] Kaur, Dr. Arvinder. "A Genetic Algorithm for Fault Based Regression Test Case Prioritization." *IJCA*, Foundation of Computer Science (FCS), https://www.ijcaonline.org/archives/volume32/number8/3925-5545.

[19] Saha, R.K.; Zhang, L.; Khurshid, S.; Perry, D.E. An information retrieval approach for regression test prioritization based on program changes. In Proceedings of the 37th International Conference on Software Engineering, Florence, Italy, 16–24 May 2015; IEEE Press: Piscataway, NJ, USA, 2015; Volume 1, pp. 268–279.

[20] Saini, R., Saini, S., Gupta, D., & Rana, A. (2012, July). Reduction in Test Cases using Regression testing approach and cost effective test prioritization testing techniques - APFD measure. International Journal of Scientific and Research Publications, 2(7), 1-8.

[21] Upadhyay, A.K.; Misra, A. Prioritizing test suites using clustering approach in software testing. Int. J. Soft Comput. Eng. IJSCE 2012, 2, 222–226.

[22] *E Of Test Case Prioritization Technique Based on Regression \Testing*.https://airccse.net/journal/ijsea/papers/5614ijsea08.pdf.

[23] P. Tonella, P. Avesani and A. Susi, "Using the Case-Based Ranking Methodology for Test Case Prioritization," *2006 22nd IEEE International Conference on Software Maintenance*, 2006, pp. 123-133, doi: 10.1109/ICSM.2006.74.

[24] V. H. S. Durelli *et al.*, "Machine Learning Applied to Software Testing: A Systematic Mapping Study," in *IEEE Transactions on Reliability*, vol. 68, no. 3, pp. 1189-1212, Sept. 2019, doi: 10.1109/TR.2019.2892517.

[25] Xu, Z.; Gao, K.; Khoshgoftaar, T.M. Application of fuzzy expert system in test case selection for system regression test. In Proceedings of the IRI-2005 IEEE International Conference on Information Reuse and Integration, Las Vegas, NV, USA, 15–17 August 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 120–125.