

## **ASSIGNMENT 2 (TESTING)**

**Name: Omkar Shrikant Gawade.**

**ID : 1001967237**

### **Machine Learning in software engineering**

#### **INTRODUCTION**

Software testing can represent anywhere from 25-40% of a project's budget. It is an expensive process, but a necessity for ensuring software works correctly. Larger companies will have entire teams dedicated to testing, but for many smaller development teams, a large testing department is beyond the available budget. Meanwhile, market demands encourage rapid development phases, which leaves little time for adequate testing.

Companies large and small are looking for ways to reduce costs and improve the scope and reliability of testing, all while delivering software that meets client expectations and deadlines. Due to this companies are trying to find automated solutions to test softwares in an efficient way which leads to the era of automated testing.

The last decade has seen a relentless push to deliver software faster. Automated testing has emerged as one of the most important technologies for scaling DevOps, companies are investing enormous time and effort to build end-to-end software delivery pipelines, and containers and their ecosystem are holding up on their early promise. Software testing is an essential part of the software development lifecycle (SDLC). Initially, testing was done manually-a process that took a lot of time and effort to execute. Then came test automation, which leverages software tools to run tests and identify bugs. Automation revolutionized the testing process and brought along many benefits such as faster feedback and higher test coverage. Scaling test automation and managing it over time remains a challenge for DevOps teams.

Today, machine learning (ML) and artificial intelligence (AI) have entered the software testing space, redefining a new era in the software development industry. AI in software testing aims to make testing smarter and more reliable. AI and ML have made a remarkable impact in software testing, where their implementation has made the testing process easier, faster, and more accurate.

Well-written software must meet its functional and non-functional requirements, moreover, it should also follow relevant coding guidelines. Programmers that follow coding guidelines consistently write code that avoids unnecessary complexities, furthermore, such code is easy to understand.

You can read more about the importance of coding guidelines in "Importance of code quality and coding standards in web application, mobile app and software development". How do you ensure that your team follows coding standards? Well, code review is the only way, however, that's an expensive affair! You need to ensure that reviewers focus on what matters, therefore, you need tools that can find common deviations from coding standards in software engineering. Machine learning can help here since ML-powered tools can find such common deviations. This can have a significant positive impact on your software development projects

#### **FIRST ARTICLE**

The Testing of software is an important and critical phase of the software development cycle (SDLC). The quality of software can be increased with the proper software testing. The role of software testing can be realized with the fact that testing has been established as a complete procedure rather than a single part in SDLC. This process begins in parallel to the SDLC. As soon as the requirements of a

project are finalized, the testing process can be started. The testing process similar to SDLC is also composed of a well defined sequence of steps and is known as software testing life cycle (STLC). The major benefit of STLC is to involve the testers at an early phase of implementation of the software. This has a considerable advantage in the project plan and cost.

But the problem with software testing is that there are a large number of test cases to test. It means the domain of software testing is too large to test. We can also say that keeping in mind the processing resources and time, all these test cases are not possible to execute. In other words, the complete testing is not possible. So we need to concentrate on effective testing rather than complete testing. The testing must be performed on some selected test cases covering all the aspects of software aspects. This is known as effective testing. Effective testing is to devise some methods such that selected subsets of test cases can be executed instead of all sets of test cases. The effective testing needs a proper planning keeping in all the software quality factors and the application in hand to test.

Whenever there is change in software configuration, i.e. there is some change due to any reason (bugs, addition of new functionality, etc.), there is a chance that other modules of the software may get affected. In this case, we need to test the whole software again.

So to avoid all the hassle, machine learning concepts are used. machine learning (ML) is becoming a standard part of many software systems. A trained model in your system may be surfacing predictions directly to users to help them make a human decision, or it may be making automatic decisions within the software system itself. Whether the ML in a system is developed in-house or is retrieved from a 3rd party's pre-trained model API, if production software utilizing a trained model's predictions is being used, it needs to be rigorously tested like you would any other aspect of your software.

## **SECOND ARTICLE**

With the extensive use of machine learning to develop software applications, it has become essential to follow best software practices for machine learning-based systems as well. The success of any software application is totally dependent on the effectiveness of the testing strategy to test the software applications. The system developed using machine learning required regular training of models and every time the predictions of the trained model are required to be validated and tested. The use of machine learning in software testing facilitates the software testing. For performing the testing of artificial intelligence-based systems, testing is required at each phase of the model development such as testing of data cleansing, feature engineering etc. The generation of the test cases using machine learning is easier and fastest. Machine learning also helps to design the technique to reduce the test maintenance and provide a simpler test analysis.

Various algorithms of machine learning like the support vector machines, semi-supervised K - means, K - Nearest Neighbor classification, Q-learning Support Vector Regression and Multi-layer perception are applied to generate the effective test cases and also prioritize the test cases. The testing strategies based on machine learning are capable of delivering reliable and robust software within the time and allocated budget.

## **CONCLUSION**

As you can see both views truly support using machine learning in software testing. Though implementation of machine learning in any industry is complex and time consuming, It saves a lot of time and avoids human error. But according to my view, machine learning is really a future of not only software testing but in every application area like healthcare, business predictions, complex calculations, etc. along with that we all need to keep one thing in mind is that for machine learning every problem is solvable until it's not.

## References

1. <https://ieeexplore.ieee.org/document/8705808>
2. <https://www.applause.com/blog/how-machine-learning-improves-test-automation>
3. <https://insidebigdata.com/2021/01/27/how-ai-and-machine-learning-will-shape-software-testing/>
4. <https://towardsdatascience.com/software-testing-and-machine-learning-45f014968c62>
5. <https://ieeexplore.ieee.org/document/9702427/references#references>