

# **VARIOUS SOFTWARE TEST CASES FOR COVID-19 POSITIVE CASES PREDICTION SYSTEM**

Guntipally Shubhasri<sup>1</sup> Dr. G.N.R. Prasad <sup>2</sup>

<sup>1</sup> MCA IV Semester, Department of master of computer applications  
Chaitanya Bharathi Institute of Technology (A), Gandipet, Hyderabad – 500 075.

E-Mail : shubhasriguntipally5@gmail.com

<sup>2</sup> Sr. Asst. professor, Department of master of computer applications  
Chaitanya Bharathi Institute of Technology (A), Gandipet, Hyderabad – 500 075.

E-Mail : gnpr@cbit.ac.in

## **ABSTRACT**

In the process of developing software, testing is essential. Testing is a crucial and beneficial phase of the software development life cycle in every organisation. Software testing methodologies vary from firm to company. Testing software is not a simple process. There will be difficulties with both coding and decoding every single day. In order to enhance decision-making regarding the future course of action, machine learning (ML) based forecasting methods have demonstrated their usefulness to foresee in perioperative outcomes. The detection and prioritisation of negative aspects for a threat was a long-standing requirement in many application domains where ML models were deployed. Many prediction techniques are now in use to address forecasting issues. This study demonstrates how machine learning (ML) models may be used to forecast the number of COVID-19 patients who will be affected in the future, which is currently regarded to be a threat to mankind. Four widely used forecasting models, including support vector machine (SVM) and linear regression (LR), were used in this study to anticipate the potentially dangerous ingredients (SVM).

Keywords : Software Testing, Machine learning, Covid-19, Linear Regression, Support vector machine.

## **1. INTRODUCTION**

Over the past ten years, machine learning (ML) has established itself as a significant subject of research by finding solutions to several extremely challenging and complicated practical issues. Nearly all of the real-world domains were represented in the application fields, including healthcare, AV (autonomous vehicles), business applications, NLP (natural language processing), intelligent robotics, gaming, climate modelling, speech, and image processing. In contrast to traditional algorithms, which execute programming instructions based on conditional statements like if-else, ML algorithms learn generally by trial and error.

The two basic categories into which an examination of software quality may be divided are static and dynamic analysis. The inspection of various documents, such as source code, software models, design papers, and requirements documents, is the basis of static analysis. Some of the fundamental techniques used in static analysis include code review, inspection, walkthrough, algorithm analysis, and correctness checking. It doesn't actually involve executing the

developing code. Dynamic analysis of a software system involves actual programme execution to find possible programming faults. Performance and behavioural traits of the programme are also addressed. Programs can be run using both standard and hand-selected input values. The input set for a software might frequently be too large. The likelihood of a system failure will be lower, the cost of testing will be lower, and the system will work well or not work properly. These are the testing's primary objectives.

## **2. REVIEW OF LITEATURE**

The fact that a person might have the virus for several days without exhibiting symptoms makes its propagation the most difficult. Nearly all of the nations have imposed either partial or strict lockdowns across the impacted regions and cities due to the causes of its spread and the risk it poses. Currently, medical experts from all around the world are working to find a suitable vaccination and treatment options for the illness. Since there is currently no recognised treatment to eradicate the virus, all national governments are concentrating on preventative measures that can halt its spread. The most crucial safety measure is "be informed" about any COVID-19-related information. Numerous researchers are investigating the various facets of the pandemic to add to this knowledge and provide the findings that will benefit mankind.

Since there is currently no recognised treatment to eradicate the virus, all national governments are concentrating on preventative measures that can halt its spread. This work illustrates the capacity of machine learning (ML) models to predict the number of forthcoming COVID-19 patients impacted, which is now thought to pose a threat to humanity. Four common forecasting models, including linear regression and support vector machines, were specifically employed in this work to predict the COVID-19 danger variables. Each model predicts one of three things: the number of newly infected cases, the number of fatalities, and the number of recoveries over the course of the following 10 days.

The study's findings show that applying these techniques to the present COVID-19 pandemic scenario is a potential strategy. The results show that the ES outperforms all other models, followed by LR and LASSO, which are good at predicting new confirmed cases, death rates, and recovery rates, while SVM does badly in all situations where predictions are made using the dataset that is currently available.

## **3. TESTING AND TEST CASES**

Software testing serves as the final assessment of the specification, design, and coding and is a crucial component of software quality assurance. We were motivated to plan via testing by the rising visibility of software as a system component and the expenses that come with a software failure. A programme is tested by being run with the goal of identifying any errors. As difficult as the initial design of the product can be, creating tests for software and other manufactured items can be.

There of basically two types of testing approaches.

### **3.1 BLACK-BOX TESTING**

Tests can be carried out to show a product is performing each function to the full extent for which it has been designed.

### **3.2 WHITE-BOX TESTING**

Testing can be done to determine whether the product's internal operation meets specifications and that all internal components have had sufficient use after learning about its internal workings.

This package has been tested using both white box and black box techniques. The border and intermediate conditions of the complete loop construct have been examined. The test data was created with the intention of verifying that all assumptions and reasonable choices had been made. Through the usage of exception handlers, error management has been handled.

### **3.3 TESTING STRATEGIES**

A series of operations known as testing can be organised in advance and carried out in a methodical manner. A software testing plan must take into account both high-level tests that validate key system functionalities against client requirements and low-level tests that are required to confirm that a tiny source code segment has been appropriately implemented.

One component of verification and validation is software testing. Verification is the collection of procedures used to confirm that software carries out a certain function as intended. The term "validation" refers to a variety of procedures that make sure the developed software can be linked to the specifications of the client.

Software testing's primary goal is to find errors. Unit, integration, validation, and system tests are developed and carried out in order to achieve this goal. A number of systematic test techniques that aid in the design of test cases are used to complete each test stage. The level of abstraction at which software is thought of increases with each testing phase.

The only method to guarantee the quality of software is through testing, which is an overarching activity rather than a distinct stage. The software effort should be carried out concurrently with this activity, which has its own phases of analysis, design, implementation, execution, and maintenance.

#### **3.3.1 UNIT TESTING**

Instead of delving into specifics at the statement level, this testing method treats a module as a single unit and checks the unit at interfaces and communications with other modules. In this case, the module will be viewed as a black box that accepts input and produces output. The module generates outputs based on a pre-calculated set of input combinations.

#### **3.3.2 SYSTEM TESTING**

Here, all of the individually tested individual modules will be put together to form the larger system, and system-level tests are done to ensure that all modules are operating in synchrony with one another. This testing methodology aids in ensuring that all modules that operate flawlessly when examined independently also operate cohesively with one another. In order to

ensure that no path leads to pandemonium during testing, we first generated test combinations of test paths throughout the system and created test cases to evaluate all modules at once.

### **3.3.3 INTEGRATED TESTING**

A significant quality control tool used in software development is testing. Error detection is its primary purpose. When coupled, sub functions might not give the results that are wanted. The issues can be modelled using global data structures. Integrated testing is a methodical approach for creating the programme structure while the tests are being run. The goal is to create unit test modules and build a programme structure that has been detected by design in order to find interface issues. A non-incremental integration combines all the modules beforehand and tests the application as a whole. In this case, an endless loop function will display errors. In incremental testing, the programme is built and tested in manageable chunks, allowing for the isolation and correction of any flaws. Top-down integration, bottom-up integration, and regression testing are three different incremental integration methodologies.

### **3.3.4 REGRESSION TESTING**

As the software evolves, a new module is frequently added as part of integration. Regression testing is a technique that aids in preventing the introduction of undesired behaviour or additional mistakes as a result of changes.

A selection of all test cases may be run manually during regression testing, or the software developer may use automated capture and playback tools to record the test case and results for later playback and compression. The regression suit includes various test case classes.

## **4.0 TEST CASES**

<b>S.NO</b>	<b>INPUT</b>	<b>OUTPUT</b>	<b>RESULT</b>
<b>Test Case 1 (Unit testing of Dataset)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases.
<b>Test Case 2 (Unit testing of Accuracy)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases using SVM got accuracy up to 84%. It helps for accuracy in Covid -19Future forecasting.

<b>Test Case 3 (Unit testing of Machine Learning Algorithms)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases using SVM got accuracy up to 84%. It helps for accuracy in Covid -19 Future forecasting using ML Algorithms.
<b>Test Case 4 (Integration testing of Dataset)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases.
<b>Test Case 5 (Big Bang testing)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases using SVM got accuracy up to 84%. It helps for accuracy in Covid -19Future forecasting.
<b>Test Case 6 (Data Flow Testing)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases using SVM got accuracy up to 84%. It helps for accuracy in Covid -19 Future forecasting.
<b>Test Case 7 (User interface Testing)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases.
<b>Test Case 8 (User interface Testing-Event based)</b>	The user gives the input in the form of Upload Covid-19 Dataset details.	An output is prediction of confirmed, recovered and death cases.	A result is predict confirmed, recovered and death cases using SVM got accuracy up to 84%. It helps for accuracy in Covid -19Future forecasting using ML Algorithms..

## CONCLUSION

Software development life-cycle is a structure imposed on the development of a software product. There are different activities involved in SDLC such as requirement, specification, architecture, software-construction, design, software testing, debugging, deployment and maintenance. Software testing plays a vital role in each and every phase of SDLC [11]. The goal of software tests is the same whether they are unit tests, integration tests, system tests, or acceptance tests. There is no evidence to support the claim that the system is flawless following the testing procedure. The various teams have been assigned varied duties. For instance, a different team is in charge of system testing, while component developers are in charge of component testing. Similarly, interface testing is carried out to detect errors in the interfaces of complicated components. By supporting automatic testing software solutions, test automation lowers testing expenses. For conducting testing of a software product, there are many automated testing tools available in the market and some of them I have given below. The most commonly used tools for software testing are: Selenium, TestingWhiz, HPE Unified Functional Testing, Watir and Katalon Studio. But the manual testing is very important in the machine learning algorithm.

## REFERENCES

- 1.S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PloS one*, vol. 13, no. 3, 2018.
2. G. Bontempi, S. B. Taieb, and Y.-A. Le Borgne, "Machine learning strategies for time series forecasting," in *European business intelligence summer school*. Springer, 2012, pp. 62–77.
3. F. E. Harrell Jr, K. L. Lee, D. B. Matchar, and T. A. Reichert, "Regression models for prognostic prediction: advantages, problems, and suggested solutions." *Cancer treatment reports*, vol. 69, no. 10, pp. 1071–1077, 1985.
- 4.P. Lapuerta, S. P. Azen, and L. LaBree, "Use of neural networks in predicting the risk of coronary artery disease," *Computers and Biomedical Research*, vol. 28, no. 1, pp. 38–52, 1995.
- 5.K. M. Anderson, P. M. Odell, P. W. Wilson, and W. B. Kannel, "Cardiovascular disease risk profiles," *American heart journal*, vol. 121, no. 1, pp. 293–298, 1991.
6. H. Asri, H. Mousannif, H. Al Moatassime, and T. Noel, "Using machine learning algorithms for breast cancer risk prediction and diagnosis," *Procedia Computer Science*, vol. 83, pp. 1064–1069, 2016.
7. F. Petropoulos and S. Makridakis, "Forecasting the novel coronavirus covid-19," *Plos one*, vol. 15, no. 3, p. e0231236, 2020.
- 8.G. Grasselli, A. Pesenti, and M. Cecconi, "Critical care utilization for the covid-19 outbreak in lombardy, italy: early experience and forecast during an emergency response," *Jama*, 2020.
9. WHO. Naming the coronavirus disease (covid-19) and the virus that causes it. [Online]. Available: [https://www.who.int/emergencies/diseases/novelcoronavirus-2019/technical-guidance/naming-the-coronavirus-disease-\(covid-2019\)-and-the-virus-that-causes-it](https://www.who.int/emergencies/diseases/novelcoronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it)
- 10.C. P. E. R. E. Novel et al., "The epidemiological characteristics of an outbreak of 2019 novel coronavirus diseases (covid-19) in china," *Zhonghua liu xing bing xue za zhi= Zhonghua liuxingbingxue zazhi*, vol. 41, no. 2, p. 145, 2020.
11. Involve testing throughout the SDLC available at <http://www.silverpath.com>