NATURALISTA CAMPANO
ISSN: 1827-7160
Volume 28 Issue 1, 2024

Analysis of Strength and Capabilities of Major Machine Learning Algorithms Used in Software Testing and Quality Assurance

Dr. Sunil Khilari[1], Dr. Balasaheb Bhamango[2], Dr. Tanaji Dabade[3], Prof. Ravi Kale[4], Prof. Asmita Hendre[5], Dr. Zarina Shaikh[6]

[1,2,3,4,5]Navsahyadri Group of Institutions, Faculty of Management, Pune.
[6]Poona Institute of Management Sciences and Entrepreneurship, Pune.

Abstract: As an Information Technology researchers , it is significant that we have a understand and analyze what's going on "under the cover" , when utilizing easy-to apply Machine Learning Algorithms, libraries, rather than simply plugging and keep going through fit-predict software quality. his research study take an descriptive analytics of the major machine learning algorithms and their strength and capabilities and all the applications and deployment tools that make it easier to present software project metrics to end-users of both Software Quality Experts and Leads .Here researcher is studying how to extract hidden insight from data and use for analyzing software project metrics Further researcher does the analysis of various Machine learning algorithms used for prioritising algorithm aspects for software project metrics, such as code complexity, predict project completion date and developer productivity etc.

Keywords: Analysis, Strength, Capabilities, Machine Learning Algorithms, Software Testing, Quality Assurance.

1. Introduction

a) Software Testing

Software testing and Quality Assurance (QA) are two related but distinct processes within the software development lifecycle. Software testing is the process of identifying and verifying that software applications or programs will meet user requirements, and quality assurance is the process of ensuring that software meets established quality standards. Both processes are necessary to deliver a high quality product. Software testing focuses on evaluating a product or service to determine whether it meets design specifications and meets user needs. Quality assurance, on the other hand, is a broader term focusing on the overall process of ensuring that a product or service meets design specifications and meets user needs. Quality assurance is a continuous process that begins during the planning stages of a product or service and continues through the development, testing, and deployment of the product or service. Software testing involves creating test plans, writing test cases, and performing manual and automated tests to ensure that the product meets specified requirements. Software testing is the process of running a program or system with the intention of finding defects or errors. It is performed to verify the functionality of the software product and ensure that it meets the design requirements and specifications.

Table1 Identification s/w testing features and restrictions w.r.t benefits

Characteristics of Software Testing	**Advantages of Software Testing**	**Limitations of Software Testing**
Purpose:	Enhances Quality	Time
Scope	Early Detection	Assumptions
Involvement	Cost Saving	Scope
Techniques	Improved User Experience	Complexity
Deliverables	Securing software code **(T. Menzies,2010)**	Less Involvement of Testers
Responsibilities	Dvelop test cases	Difficult to Make a Product Error-Free
Define the test environment	Write scripts	Impact on code efficiency

Analyse test results	Submit defect reports	Large number of human resources

With the help of extensive literature review above characteristics and respective advantages and challenges related to software testing is identified.

b) Quality Assurance

Quality Assurance (QA) is the process of ensuring that a software product meets specified quality standards. It is a continuous process that includes planning, designing, developing and testing software. QA focuses on the quality of a software product and is performed by software engineers and developers. Its primary objective is to ensure that the software meets the customer's requirements and is of high quality. QA is a process-oriented approach that involves implementing best practices and improving processes to ensure software quality. It includes activities such as requirements gathering, design review, code review, unit testing, system testing, integration testing, and regression testing.

Table-2 Identification s/w QA features and restrictions w.r.t benefits

Characteristics of Software Quality Assurance	**Advantages of Software Quality Assurance**	**Limitations of Software Quality Assurance**	**Challenges in Software Quality Assurance**
Involvement	Reliability	Knowledge Limitation	Poor motivation
Techniques	Customer Satisfaction	Human Error	Knowledge gap
Responsibilities	Enhances Usability	High initial investment in resources	Lack of teamwork
Deliverables	Increased Productivity	Process Limitation:	Complex systems
Scope	Enhancing Brand Reputation	Difficult to measure software qualities	Inadequate support
Portability	Ensuring Compliance with Regulatory Requirements	Difficult to determine quality measurement metric	Organisational rigidity.
Interoperability	Holding formal technical reviews	High variance on objectives	Changing requirements
Security	Implementing a multi-testing strategy	Reworking/remaking products	Tight deadlines

With the help of extensive literature review above characteristics and respective advantages and challenges related to software Quality Assurance (QA) is identified.

Application of Quality Assurance

- **Process Auditing:** Process auditing is a type of quality assurance that verifies the processes of a system. This is done to ensure that the processes of the system are efficient and effective.
- **Automated Testing:** Automated testing is a type of quality assurance that verifies system functionality using automated tools. This is done to ensure that the system is working as per the specified requirements.
- **Defect Tracking:** Defect tracking is a type of quality assurance that tracks and manages defects in a system. This is done to ensure that all faults in the system are identified and resolved.
- **Configuration Management:** Configuration management is a type of quality assurance that verifies and manages system configuration. It is performed to ensure that the system is configured according to the specified requirements.
- **Risk Management:** Risk management is a type of quality assurance that verifies and manages the risks associated with a system. This is done to ensure that the system is safe and protected against any potential risks.
- **Design and code review:** Design and code reviews are a type of quality assurance that validates the design and code of a system. This is done to ensure that the design and code of the system meet the specified requirements and are of high quality.

Machine Learning Tasks

A machine learning task is the type of prediction (**F. Yucalar,2020**) or inference being made, based on the problem or question that is being asked, and the available data. For example, the classification task assigns data to categories, and the clustering task group’s data according to similarity

Objectives of Study

Following are some objectives are established to work on gap identified for this research work as under-

1. To identify various machine learning algorithms used in Software Testing and Quality Assurance
2. To identify strength and capabilities of machine learning algorithms with respect to software project metrics

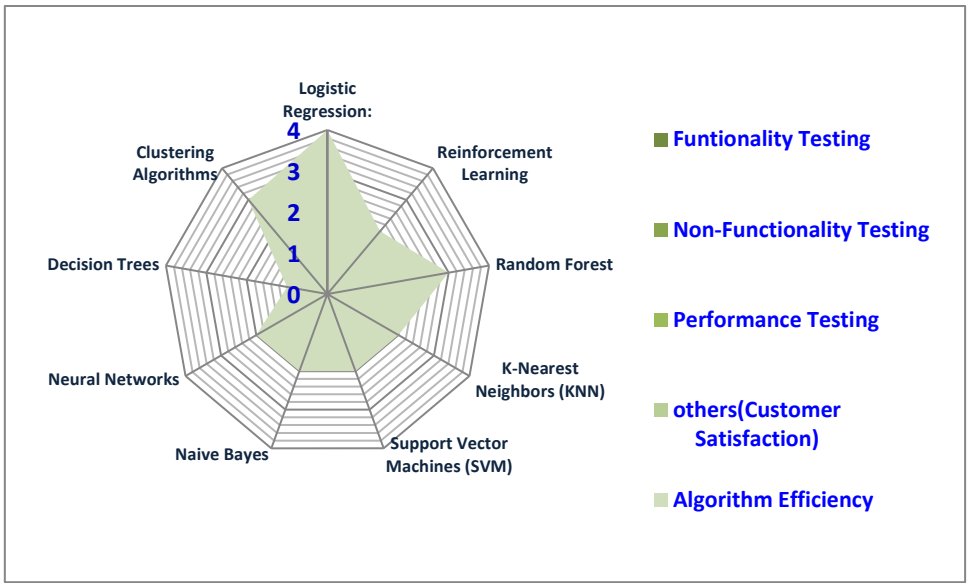
Data Collection

Objective-1 - To identify various machine learning algorithms used in Software Testing and Quality Assurance Machine learning (ML) algorithms are increasingly being applied in software testing and quality assurance to automate testing processes, detect defects, improve test coverage, and optimize testing strategies. Here are several machine learning algorithms commonly used in software testing and quality assurance **(Y.Mahmood, 2021)**

Table-3 Analysis of ML algorithms and its area of usage in s/w testing and QA

Machine Learning Algorithm	Area of Usage –s/w testing and QA			
	Functionality Testing	Non-Functionality Testing	Performance Testing	Others(Customer satisfaction)
Reinforcement Learning	√	×	√	√
K-Nearest Neighbors (KNN)	√	×	×	√
Decision Trees	×	×	√	×
Support Vector Machines (SVM)	√	×	√	×
Random Forest	×	√	√	√
Naive Bayes	√	×	√	×
Logistic Regression:	√	√	√	√
Clustering Algorithms	×	×	√	×
Neural Networks	√	×	√	×

Based on live working project above machine learning algorithms are analysed with s/w testing and QA aspects.



Graph-1 Analysed and priories ML algorithms w.r.t. s/w testing and QA

Above graph indicate that logistic regression is the best for major software project testing consequently clustering algorithms, reinforcement learning, decision tree, and random forest comes sequentially.

Rank-1: Random Forest:
Random Forest is an ensemble learning technique that uses multiple decision trees to improve prediction accuracy. In software testing, Random Forest can be used for defect prediction, test case prioritization, and identifying relevant features for test automation **(D. Bowes, T. Hall, and J. Petrić,2018)**

Rank-2: Logistic Regression:
Logistic Regression is commonly used for binary classification tasks, such as classifying defects as either high or low severity. It can also be used for estimating the probability of defects occurring in specific software components.

Rank-3; Clustering Algorithms:
Clustering algorithms such as K-Means, DBSCAN, and hierarchical clustering can be used for grouping similar software components, test cases, or defect reports. Clustering can help identify patterns, outliers, and relationships within software artifacts, facilitating quality assurance tasks.
By leveraging these machine learning algorithms, software testing and quality assurance teams can enhance their efficiency, effectiveness, and accuracy in identifying defects, improving software quality, and optimizing testing processes **(R. Malhotra,2012)**

Objectives-2- To identify strength and capabilities of machine learning algorithms with respect to software project metrics
Machine learning algorithms offer various strengths and capabilities when it comes to analysing software project metrics. Here are some key strengths and capabilities of machine learning algorithms in this context **(Vijay, T. John, D. M. G. Chand, and D. H. Done.,2017).Researcher has carried out extensive literature review and observations made on working live software project and below analysis is made as-**

Table-4 Analysis of ML algorithms w.r.t. testing metrics

Algorithm	Strength	Weakness	Testing metrics
Reinforcement Learning	• Predictive Analytics: • Anomaly Detection • Feature Selection and Engineering: • Classification and Regression	• More data required • Not suitable to use for solving simple problems • To learn from trial and error, • Costly and time-consuming.	• Unit Testing • Data Testing • Cross-Validation • Performance Testing
K-Nearest Neighbors (KNN)	• Time-Series Analysis: • Pattern Recognition • Pattern Recognition: • Optimization and Decision Support:	• It does not create a generalized separable model • There is no summary equations • Slow speed • Memory and storage issues for large datasets • Sensitivity to the choice of k and the distance metric	• Functionality testing • Load testing • Stress testing • Integration Testing
Decision Trees	• Interpretability • Less Data Preparation • Non-Parametric • Versatility • Non-Linearity	• Over fitting • Feature Reduction • Data Resampling • Optimization	• Regression Testing • Performance Testing • Security Testing • User Acceptance Testing
Support Vector Machines (SVM)	• Margin of separation between classes • High dimensional spaces. • Memory efficient	• Not suitable for large data sets • Not support if classes are overlapping.	• System Testing • Functional Testing • Acceptance Testing

	• Unstructured and semi structured data	• Not suitable for features data point are more than training data samples • Long training time for large dataset	• Smoke Testing
Random Forest	• To improve the accuracy • Flexible to both classification and regression problems • Support both categorical and continuous values. • Automates missing values present in the data.	• Not easily interpretable. • Does not provide complete visibility into the coefficients as linear regression • Over fitting • More Training time • Complexity:	• Non-functional testing • Functional Testing • Acceptance Testing • Smoke Testing
Naive Bayes	• Easy to implement and understand(Okutan,2012) • It doesn't require as much training data • It is highly scalable with the number of predictors and data points (N. Bouguila,2008) • It is fast and can be used to make real-time predictions	• Zero-frequency problem • All predictors are assumed to be independent, • Issue in usability in real-world scenarios • its estimations can be off in some instances (E. Rashid,2012)	• Usability Testing • Compatibility Testing • Scalability Testing • Stability Testing
Logistic Regression:	• Better when the data is linearly separable. • highly interpretable • It does not require tuning. • IT help to identify data anomalies,	• fails to predict a continuous outcome • assumes linearity between the dependent variable and the independent variables • Not be accurate if the sample size is too small. • It constructs linear boundaries	• Acceptance Testing • Smoke Testing • stress testing • Integration Testing
Clustering Algorithms	• To identify obscure patterns and relationships within a data set. • It carries out exploratory data analysis. • Best for automatic recovery from failure, • It recover data without user intervention	• complexity • inability to recover from database corruption • determining the optimal number of clusters (K) • limited adaptability to categorical data	• Sanity Testing • Exploratory Testing • Adhoc Testing • Globalization Testing
Neural Networks	• Ability to handle complex data • Non-linear modelling capabilities • Adaptability and learning capabilities • Feature extraction capabilities	• Need for large amounts of labelled training data • Computationally intensive and resource-consuming • Black box nature can hinder interpretability • Complexity and difficulty in model tuning	• Security Testing • User Acceptance Testing • Performance Testing • Adhoc Testing

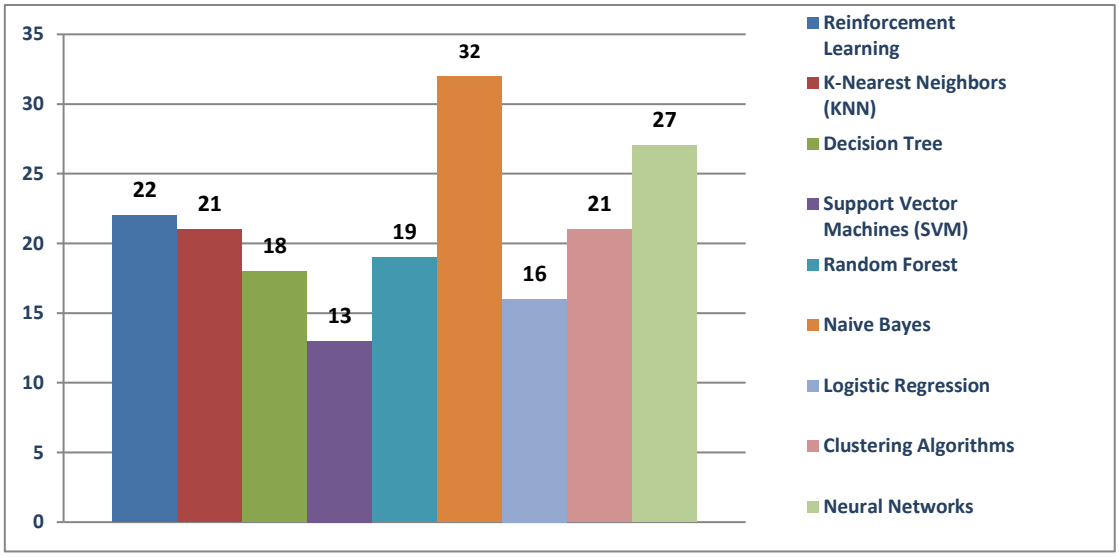
Data Analysis and Interpretation

Based on experiments and observation on live software projects with respect to algorithm key aspects applicable to software testing and quality assurance has analysed as below0

Table-5 Analysis of ML Algorithms w.r.t characteristics.

Aspects of Algorithms	Reinforcement Learning	K-Nearest Neighbors (KNN)	Decision Tree	Support Vector Machines (SVM)	Random Forest	Naive Bayes	Logistic Regression	Clustering Algorithms	Neural Networks
Memory	3	2	1	2	1	3	1	2	3
Structure	2	2	3	2	1	3	1	2	3
Complexity	3	2	1	1	1	2	2	1	2
Flexibility	1	2	2	1	2	3	1	2	3
Learning Mechanism	2	2	1	1	2	3	1	2	1
Speed	2	2	1	1	2	3	1	1	2
Scalability	3	2	1	1	2	3	1	2	3
Replication and Control	1	2	2	1	2	3	2	2	2
Processing capability	1	2	2	1	2	3	1	3	3
Interpretability	3	1	3	1	2	3	3	3	3
Fault Tolerance	1	2	1	1	2	3	2	1	2

1-Low, 2–Medium, 3-High



Graph-2 Representation of total aspect levels of ML Algorithms

Level Total-->

Above graph indicates that Support Vector Machines (SVM) algorithm is best for software testing and QA project matric as compared with major ML algorithms whereas each algorithm has their individual strength and weakness.

Machine learning algorithms can help identify relevant features or variables from a large pool of software project metrics (S. Jha, 2019). Feature selection techniques, such as recursive feature elimination or feature importance analysis, can prioritize metrics that have the most significant impact on project outcomes, enabling more focused analysis and decision-making. Classification and regression algorithms can analyse software project metrics to categorize projects into different classes or predict **(X. Wang, Y. Zhang, L. Zhang and Y. Shi ,2010)** continuous variables. For instance, classification algorithms can classify projects as high or low risk

based on metrics like code complexity and defect density, while regression algorithms can estimate project effort or quality based on various project metrics. Clustering algorithms can group software projects or components based on similarities in their metrics, enabling segmentation and comparison across different groups.

3. Conclusion

In summary, each machine learning algorithm has unique strengths and capabilities that can be leveraged for various tasks in software testing and quality assurance. By understanding the characteristics of different algorithms, organizations can choose the most appropriate techniques to address specific challenges and improve the effectiveness of their testing processes.. By leveraging the strengths and capabilities of these machine learning algorithms, organizations can enhance their software testing and quality assurance processes, leading to improved software quality, reduced defects, and increased efficiency in delivering reliable software products. However, it's essential to choose the appropriate algorithms based on the specific requirements and characteristics of the software projects to achieve the desired outcomes effectively. logistic regression is the best for major software project testing consequently clustering algorithms, reinforcement learning, decision tree, and random forest comes sequentially. Support Vector Machines (SVM) algorithm is best for software testing and QA project matric as compared with major ML algorithms whereas each algorithm has their individual strength and weakness.

Machine learning algorithms can optimize software development processes by identifying opportunities for improvement and providing decision support to project managers and stakeholders. Optimization algorithms, such as genetic algorithms or reinforcement learning, can optimize resource allocation, scheduling, and task prioritization based on project metrics and objectives. Machine learning algorithms can recognize complex patterns and relationships within software project metrics that may not be apparent through traditional analysis methods. Pattern recognition algorithms, such as neural networks or decision trees, can identify correlations between metrics, detect recurring patterns in project data, and uncover hidden insights that inform process improvements and risk mitigation strategies.

Overall, machine learning algorithms offer powerful tools for analysing software project metrics, enabling organizations to gain valuable insights, make data-driven decisions, and improve the efficiency and quality of software development processes.

4. References

1. S. Jha et al.Deep Learning Approach for Software Maintainability Metrics Prediction,IEEE Access,(2019)
2. E. Rashid et al.Software Quality Estimation using Machine Learning: Case-based Reasoning Technique,Int. J. Comput. Appl.(2012)
3. T. Menzies et al.Defect prediction from static code features: current results, limitations, new approaches Automated Softw. Eng.(2010)
4. F. Yucalar et al.Multiple-classifiers in software quality engineering: Combining predictors to improve software fault prediction ability,Eng. Sci. Technol. Int. J.(2020)
5. Y.Mahmood et al.Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation,Softw. Pract. Experience,(2021)
6. Okutan et al.Software defect prediction using Bayesian networks,Empirical Softw. Eng.(2012)
7. R. Malhotra et al.Fault Prediction Using Statistical and Machine Learning Methods for Improving Software Quality, J. Inf. Process. Syst.(2012)
8. N. Bouguila et al.A bayesian approach for software quality prediction,IEEE,(2008)
9. Software Quality Measurement Analysis based on Techniques, Criteria, Metrics, Models, and Datasets,2022, 2022 8th International Conference on Contemporary Information Technology and Mathematics, ICCITM 2022
10. 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.
11. Prediction of Software Project Quality Based on Test Measures Via Ai Methods,2023, SSRN
12. Engineering Challenges in the Development of Artificial Intelligence and Machine Learning Software Systems,2023, System Reliability and Security: Techniques and Methodologies

13. 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.
14. 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.
15. https://www.testingxperts.com/blog/ml-testing#Types%20of%20ML%20Testing
16. https://www.wallstreetmojo.com/artificial-neural-network/
17. https://www.javatpoint.com/types-of-software-testing