Comparative Study Between Machine Learning Algorithms Applied to Support QA Contracting Processes

David Santiago Bejarano Velandia, Jorge Enrique Rodriguez Rodriguez, and David Stevens Gonzalez Lizarazo

***Abstract*—This study comprehensively compares different machine learning algorithms to assess their applicability in Quality Assurance (QA) contracting procedures. The evaluated algorithms encompass Nearest Neighbors, Linear SVM, Radial SVM, Gaussian Process, Decision Tree, Neural Network, Logistic Regression, Naive Bayes, and QDA. Following the Knowledge Discovery Database (KDD) process, the methodology includes a diverse set of evaluation metrics such as F1-Score, recall, accuracy, and AUC-ROC, as well as learning curves, boundary maps, confusion matrices, Matthews Correlation Coefficient (MCC), and complexity curves. According to the Gartner Magic Quadrant assessment, the results suggest that Neural Network, QDA, and Gaussian Process models exhibit strong performance and thorough evaluation, making them optimal for the case study presented in the paper. In contrast, Nearest Neighbors and Linear SVM models are considered suboptimal, indicating an opportunity to explore the reasons behind their behavior in the case study and how to modify them for improved results. The other algorithms also present various possibilities for adaptation to the current case study, either as models with limited analysis or as imprecise models that can offer valuable insights for future work on optimizing them more effectively. This study significantly contributes to advancing machine learning applications in recruitment procedures.**

***Index Terms*—Machine Learning, Quality Assurance, hiring processes, algorithm comparison, bias reduction.**

I. Introduction

MACHINE learning is a subdivision of artificial intelligence focused on enabling systems to learn without explicit programming. This is achieved by allowing the system to identify patterns and anticipate future actions. Machine learning is applied in various domains, including natural language processing, genetic sequence analysis, robotics, financial risk assessment, threat detection, compiler optimization, semantic web, computer security, software engineering, and image processing, as highlighted by Panesar [1].

In computing, machine learning leverages statistical methods and algorithms to help machines enhance their performance on tasks by learning from data. According to Panesar [1], the ultimate objective is to create models that can learn from examples and training data, and subsequently apply that knowledge to new scenarios.

Manuscript received December 9, 2023; revised August 4, 2024.

David Santiago Bejarano Velandia is an undergraduate student of Telematics Engineering, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. (e-mail: dsbejaranov@udistrital.edu.co)

Jorge Enrique Rodriguez Rodriguez is an associate professor of Telematics Engineering, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. (e-mail: jerodriguezr@udistrital.edu.co).

David Stevens Gonzalez Lizarazo is an undergraduate student of Telematics Engineering, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. (e-mail: dsgonzalezl@udistrital.edu.co).

The software quality assurance industry has experienced substantial changes, resulting in a heightened demand for specialized professionals. The increasing complexity of modern software systems and the emphasis on their effectiveness and security has led to a greater need for QA professionals. As a result, hiring teams are facing challenges in finding candidates with the required skills and knowledge.

Recruiting QA professionals has become highly competitive. According to a report by the International Association of Software Testing Professionals (ISTQB), the demand for QA professionals has surged by 30% over the past three years, leading to a shortage of talent in the job market. Consequently, companies are now vying to attract the most qualified professionals. [2]

In the rapidly evolving digital landscape, machine learning has emerged as a valuable asset for hiring teams. Its capacity to process intricate data, uncover underlying trends, and drive data-driven decision-making renders machine learning algorithms well-suited for specialized and competitive hiring procedures, instilling confidence in the recruitment process.

Notably, machine learning has revolutionized the hiring landscape, particularly within the software QA industry. Through the analysis of extensive datasets and pattern recognition, machine learning algorithms can effectively pinpoint candidates whose profiles align closely with specific job prerequisites. According to Sharma [3], this technology can streamline the selection phase, ultimately alleviating the workload for hiring teams and conserving valuable time and resources.

In this paper, a thorough analysis of machine learning algorithms commonly used in software QA recruitment is presented. It delves into the core concepts of machine learning and its impact on enhancing recruitment efficiency.

This research is notable for its in-depth examination of different algorithms, their comparative analysis, and their practical application in a controlled scenario. It underscores the significance of prior research and the benefits of comparing various machine learning options for recruitment. The inclusion of examples and comparisons throughout the text reflects the intent to assess different alternatives using specific metrics.

Key sources for this research include Panesar [1] and Sharma [3], whose contributions have influenced the problem statement of this paper and advocated for the adoption of machine learning in recruitment, particularly in software QA. This paper has not been previously published.

II. Problem

The field of human resource management is constantly evolving, driven by technological advancements, shifts in work trends, and updates in regulations that influence work dynamics and employee expectations [4].

As Russo et al. [5] highlighted, this evolution is particularly evident in software quality assurance, where technical expertise and precision are imperative. The introduction of machine learning has transformed the recruitment and selection processes within this field.

The incorporation of machine learning into human resource management offers substantial advantages. Adnan [6] explains that algorithms can swiftly and accurately analyze extensive candidate data, identifying patterns and characteristics that traditional methods might overlook. This not only streamlines the selection process but also enhances the likelihood of identifying highly qualified candidates for specific roles.

Automating routine tasks in the selection process enables Human Resources (HR) professionals to concentrate on more strategic initiatives, such as employee development and retention, which are crucial for an organization's success.

However, this technological revolution also presents significant ethical challenges [6]. One pressing concern is candidate privacy, as the collection and analysis of personal data raise questions about the handling of sensitive information. Additionally, there is a risk of algorithmic bias, where algorithms may reflect inherent biases in the training data, potentially leading to unfair or discriminatory hiring decisions.

A. Competency-Based Selection and Its Relation to the Theory of Human Talent Management

In the modern field of human resource management, evaluating competencies during the hiring process is essential. This approach moves beyond the traditional emphasis on work experience and academic qualifications, in line with human talent management theory principles.

Human talent management theory emphasizes identifying and nurturing the competencies crucial to an employee's performance. It recognizes that a robust academic foundation or extensive professional experience, in isolation, may be insufficient. What truly matters are the skills, knowledge, and practical abilities that render an individual suitable for a particular role. [7]

Competency-based selection employs objective, data-driven techniques to assess candidates. Instead of relying solely on subjective interviews or personal judgments, this method measures specific competencies relevant to a given position. Approaches include skills assessments, simulated exercises, and evaluations based on past performance.

This method leads to more informed hiring decisions and ensures employees align with the organization's objectives. By choosing candidates with the right competencies, organizations enhance their ability to adapt and swiftly contribute to success.

B. Digitization and Process Automation: A Reflection of Management Theory

In the modern era, digitalization and process automation are vital for efficient management in all types of organizations. This transition allows managers to maximize efficiency and effectiveness, aligning with fundamental management principles.

A prime example of this collaboration is the utilization of Applicant Tracking Systems (ATS) in recruitment. ATSs are based on effective management principles and have become indispensable tools in the workplace.

ATSs facilitate precise and consistent data collection and analysis in hiring, leading to more efficient resource management and the elimination of repetitive tasks such as resume reviews and interview scheduling. This automation enables HR professionals to focus on more valuable activities, such as skills assessment and strategic decision-making.

Moreover, ATSs enhance communication between hiring team members and candidates, improving the candidate experience and positively impacting a company's reputation and talent retention. [7]

At a broader level, digitalization and process automation not only apply to recruitment but also extend to various administrative areas such as project management, inventory tracking, and accounting, among others. These approaches are aligned with the central goal of management theory, which is to maximize the efficiency and effectiveness of organizational operations by reducing human error, minimizing response time, and optimizing resource utilization.

C. Candidate Assessment Through Technical Tests: An Approach Supported by Human Capital Theory

Candidate assessment has undergone significant transformation in specialized fields such as technology. The traditional emphasis on academic or work history is being replaced by technical testing, which has become a central hiring process component. This shift is in line with human capital theory, which underscores the importance of investing in specific skills to enhance organizational performance and success [8].

Human capital theory emphasizes that an organization's most valuable asset is its people. Therefore, investing in specific skills and technical capabilities is crucial for maximizing employee potential and overall organizational performance. Academic qualifications or past work experience alone can no longer determine a candidate's suitability. Instead, practical skills and technical competencies are given priority.

Often supported by machine learning, technical testing has become a critical tool in candidate evaluation. According to Magazzino et al. [9], these tests assess candidates' ability to apply their knowledge in real-life scenarios, such as solving coding problems or performing specialized tasks.

The advantage of incorporating machine learning in assessment lies in its ability to evaluate correct answers and candidates' problem-solving strategies and creativity. This comprehensive assessment reassures HR professionals and hiring managers regarding the effectiveness of the process. It helps identify individuals with strong knowledge bases and high potential to contribute to the organization by addressing complex challenges and adapting to an ever-changing work environment.

D. Promoting Diversity and Inclusion in Recruitment: An Alliance with Diversity Management Theory

The promotion of diversity and inclusion has become integral to corporate recruitment. Many companies now understand that having ethnic, gender, and cultural diversity within their teams is morally right and essential for achieving business success. Studies indicate that diverse companies often outperform their competitors regarding profitability and innovation.

Diversity management theory emphasizes the importance of creating a workplace that values and respects individual differences, promoting diversity and inclusion. This approach brings about a range of perspectives and experiences that positively influence decision-making, innovation, and overall organizational performance.

Research has shown that ethnic and gender diversity is a driving force behind effective and innovative teams. Individuals with unique cultural backgrounds and experiences bring more nuanced and comprehensive problem-solving approaches, leading to innovative solutions. Furthermore, a diverse workforce enables companies to comprehend better and cater to the needs of a global and diverse market.

Nevertheless, having diversity on paper is not sufficient. Establishing an environment where everyone feels valued, respected, and empowered to make meaningful contributions is critical. This includes implementing policies and practices that promote fairness and equal opportunity in the workplace and eliminate any barriers that hinder employee participation.

E. Implications of the Application of Machine Learning and a Crucial Question

The growing use of machine learning in recruiting marks a milestone in human resource management. This technology automates tasks, reduces costs, and improves talent identification and selection efficiency. Applications like automated resume analysis and video interview scoring significantly advance how companies hire employees.

Amid this technological revolution, a fundamental question arises that deserves deep and thoughtful consideration: How can machine learning ethically and effectively support the hiring process? This involves many aspects of HR management and impacts decision-making and organizational culture.

Firstly, automation affects candidate search and selection. While algorithms can speed up the process by analyzing large data sets, it's crucial to ensure they are unbiased and do not perpetuate discriminatory biases. Fairness in candidate consideration must be a priority.

Skill assessment is another critical aspect. Machine learning can help identify candidates with the right skills, but these tests must be relevant and fair, without discrimination based on age, gender, race, or other protected factors. [10]

Automating interviews and candidate interactions also presents ethical challenges. Impersonal communication can feel cold and dehumanizing, negatively affecting the company's image and ability to attract top talent.

Lastly, automated job postings can overwhelm candidates with information, making it hard to find suitable opportunities.

Machine learning techniques revolutionize recruitment processes, especially in the software QA industry. These innovations, supported by management theories, promise to optimize talent identification. However, addressing ethical and equity concerns is essential to ensure these technologies are fair and beneficial to all parties involved.

III. Algorithm Selection

Machine learning involves computational methods that allow systems to automatically learn and enhance their performance through experience, without the need for explicit programming. These algorithms are successful in computer vision, natural language processing, fraud detection, and medicine. Mahesh [11] notes that recent progress in machine learning is due to advancements in computing power, modeling techniques, and optimization methods.

Machine learning includes various types, each focusing on different learning tasks [12]. Here are a few primary types of machine learning:

A. Supervised Learning Algorithms

Supervised learning is a widely used approach in machine learning. In this approach, models are trained using labeled examples, where each example includes input features and a corresponding output label [13]. The aim is to learn a function that accurately maps input features to output labels. Popular supervised learning algorithms include Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs).

The SVM algorithm solves linear and nonlinear classification problems by finding an optimal hyperplane that maximizes the separation between data classes. SVM is used in applications like image recognition and fraud detection in finance [14]. However, SVM can struggle with large datasets due to its computational complexity. [15]

Random forest is an ensemble technique using multiple decision trees to make predictions. Each tree is trained on a random sample of data and features. The predictions from each tree are combined to produce a final prediction. Random forest handles large datasets and irrelevant features well and is less prone to overfitting than individual decision trees. It is effective in applications like spam detection and image classification. [16]

Gradient Boosting Machines (GBM) is another ensemble technique that combines several weaker models into one robust model. Unlike Random Forests, GBM builds models sequentially, with each model fitting the residual errors of the previous one. This technique performs well on regression and classification problems and is popular in data science competitions [17]. However, due to its sequential nature, GBM can be slower.

The human brain inspires ANNs, which consist of multiple layers of interconnected artificial neurons. These networks can learn and extract complex features from data. ANNs excel in speech recognition, computer vision, and natural language processing [18]. However, they often require large amounts of data to train effectively and can be more difficult to interpret than other algorithms.

B. Unsupervised Learning Algorithms

Unsupervised learning involves unlabeled data; no output labels are provided during training. According to Abou [19],

unsupervised learning algorithms are designed to uncover hidden patterns or structures within the data. This is achieved through methodologies such as clustering, which involves grouping data based on similarities. One of the most commonly used unsupervised learning algorithms is K-Means.

K-Means is a widely used clustering algorithm that segments a dataset into k groups to minimize the variance within each cluster. It accomplishes this by assigning each data point to the nearest centroid and then adjusting the centroids to enhance clustering accuracy. Hastie et al. [20] highlight that K-Means is computationally efficient and performs well with high-dimensional data. Still, it is sensitive to initial centroids and requires the specification of the number of clusters.

Another clustering algorithm, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), identifies clusters based on point density. Unlike K-Means, DBSCAN does not require a pre-specified number of clusters. It effectively identifies clusters of various shapes and sizes and handles noise and outliers. However, DBSCAN may encounter challenges with high-dimensional data and is sensitive to parameter choices. [21]

Principal Component Analysis (PCA) is a commonly utilized technique for reducing dimensionality in machine learning. As Géron [22] explains, PCA identifies the principal components or directions of maximum variance in a dataset, facilitating the reduction of the dataset's dimensionality. PCA is valuable for data visualization, compression, and feature elimination. Still, it may not be suitable for nonlinear data and can lead to the loss of detailed information from the original data.

Autoencoders are a type of unsupervised learning algorithm used for feature reconstruction and extraction. These neural networks consist of an input layer, one or more hidden layers, and an output layer. The main goal of autoencoders is to reconstruct input data through a latent representation in the hidden layer. Tan et al. [23] suggest that autoencoders are helpful for dimensionality reduction, data generation, and anomaly detection, but their effectiveness depends on the architecture and training data quality.

Association algorithms, like Apriori, identify patterns or association rules in transactional or item-based datasets. These algorithms find item sets that frequently occur together. They are widely used in shopping cart analysis, product recommendations, and consumer behavior analysis. However, they can be computationally expensive and generate many trivial or useless rules. [24]

The machine learning algorithms mentioned offer different approaches for specific problems. Clustering techniques like K-Means and DBSCAN focus on grouping data, while PCA and autoencoders are used for dimensionality reduction. Association algorithms find patterns in data. The choice of algorithm depends on the problem, data characteristics, and analysis objectives. [25]

C. Algorithm Choice

The selection of supervised learning algorithms is a critical aspect of the QA recruitment process, especially for tackling classification problems. In supervised learning, a labeled dataset comprises historical instances of candidates categorized as either suitable or unsuitable for the job. This enables algorithms to discern patterns and connections from past cases and apply this knowledge to assess new candidates. The primary objective is to train the model to extrapolate from labeled data and employ this information to evaluate new candidates.

1) Nearest Neighbors: It is a fundamental machine learning tool characterized by its simplicity and effectiveness. Used in both classification and regression tasks, its operation is based on identifying a predefined number of training samples closest in the distance to a new point and then predicting the corresponding label based on these samples. This nonparametric approach is distinguished by its ability to adapt to various types of data by avoiding significant assumptions about the shape of the assignment function. Nearest Neighbors finds frequent applications in recommender systems, anomaly detection, and disease classification, such as lung cancer [26], where its versatility and robustness make it a preferred choice for tackling a wide range of problems.

2) Linear SVM: The Linear Support Vector Machine (Linear SVM) is a type of SVM with a linear decision boundary. It finds the optimal hyperplane that best separates the classes in the feature space. Due to its simplicity and effectiveness, Linear SVM is widely used in various machine learning applications. Linear SVM is known for its efficient generalization, making it ideal for high-dimensional data. However, training Linear SVM can be time-consuming, especially during the cross-validation phase for parameter selection. Methods like early stopping in iterative learning have been proposed to speed the training and maintain performance without prolonging training. [27]

3) Radial SVM: The Radial Support Vector Machine (Radial SVM) is a robust supervised machine learning algorithm that excels in classification and regression tasks. Its methodology involves identifying the optimal hyperplane to distinguish between different classes within a given feature space. This technique has been utilized across various industries, from cybersecurity to medical diagnosis, and has proven highly effective in handling complex datasets with high-dimensional data. SVMs are particularly adept in identifying clear separation margins between classes, making them an invaluable tool in analyzing challenging datasets.

4) Gaussian Process: These are essential for problems with continuous input and finite output spaces. These models extend multivariate Gaussian random variables to infinite index sets, providing versatility and computational efficiency [28]. Using Gaussian processes in a Bayesian framework allows for precise uncertainty calculations in predictions and supports standard model selection techniques. However, their scalability is challenging and can be addressed using sparse approximation techniques. [28]

5) Decision Trees: The integration of Decision Trees is supported by various technical reasons. Primarily, these algorithms are exceptionally interpretable, which is fundamental in QA hiring, as comprehending and rationalizing hiring decisions is crucial. Additionally, decision trees can proficiently handle both numeric and categorical data, which is typical in candidate datasets that may comprise diverse attributes. Furthermore, their proficiency in handling missing data and their resilience to noise in the data make them highly adaptable in real-world scenarios. They can also be seamlessly regularized to prevent overfitting, a common

concern in machine learning. [29]

6) Neural Network: Implementing machine learning through Neural Network mimics the structure and behavior of the human brain to learn from data and make predictions. This technique is essential in machine learning and intense learning and has advanced various fields like telecommunications, computer vision, and biomedical image processing. Recent studies compare traditional machine learning models with neural network-based models for tasks such as identifying malware families on Android devices [30]. Neural Network effectively tackle complex challenges by identifying and learning patterns and relationships in data.

7) Logistic Regression: It is a statistical technique used in machine learning for binary classification tasks. It aims to predict the likelihood of a specific event by fitting data to a logistic function. This method is widely used in various domains, such as disease prognosis, fake news detection, and data categorization. Logistic Regression is effective as a supervised learning algorithm, especially when the dependent variable is categorical. Its importance in machine learning comes from its simplicity and clear results. [31]

8) Naive Bayes: It is an optimal choice due to its proficiency in managing high-dimensional data and computational speed. Many candidate features are collected during the QA recruitment process, creating large data sets. Naive Bayes is advantageous as it presumes conditional independence among features, making it reliable and efficient in overcoming the curse of dimensionality. Furthermore, it is advantageous in handling categorical data or text frequently encountered during candidate evaluation.

9) QDA: Quadratic Discriminant Analysis (QDA) is a valuable classification algorithm in machine learning, especially when classes have different covariance matrices. It calculates the likelihood of a data point belonging to a class, assuming features are usually distributed within each class, and uses Bayes' theorem to make predictions. QDA has shown versatility and effectiveness in various classification and prediction tasks in machine learning applications.

The selection of algorithms, including Nearest Neighbors, Linear SVM, Radial SVM, Gaussian Process, Decision Tree, Neural Network, Logistic Regression, Naive Bayes, and QDA, was carefully made based on their appropriateness for handling various classification tasks. Each algorithm possesses unique features that make it suitable for specific scenarios. The decision to use these algorithms was based on their successful track record in numerous contexts and their ability to adjust to the particular attributes of the classification problem during the QA recruitment process.

IV. Data Preparation

The Knowledge Discovery in Databases (KDD) methodology is often used to solve complex problems to facilitate scientific research. Comendador et al. [32] noted that KDD is a structured process for extracting valuable insights from large datasets. This method has proven effective in various fields, including artificial intelligence, data mining, and data analytics.

This study rigorously applies the KDD methodology, following Fayyad et al.'s [33] steps to ensure quality and validity. Following Chen et al.'s [34] work on data mining's business applications, a comprehensive literature review establishes the theoretical foundation and research goals.

The initial phase involves selecting data by interviewing two organizations. Over the past two years, meetings have been held with candidates for the role of QA analyst in each organization to create a substantial database for the next stages of the methodology. During these sessions, candidates provide basic information, experience, QA knowledge, soft skills, and job expectations.

The candidate's interview responses are evaluated by QA expert interviewers who assign scores to specific questions based on their perception of the candidate's accuracy. This, along with other gathered information, results in 42 categorical data points per candidate in the first dataset and 38 in the second, all stored in a plain text file for traceability and future processing.

The next step involves processing and standardizing the data to identify the point of maximum equality, facilitating machine learning processes and streamlining workflow for optimal outcomes.

One effective data transformation technique involves converting open-ended responses into a standardized format. For instance, questions with positive or negative answers are assigned binary values: 'no/negative' responses are 0, and 'yes/positive' responses are 1, simplifying the data to two possible values for each dataset. Furthermore, comparable scoring systems were established, such as a scale from 1 to 5 for 'Knowledge of agile methodologies' and from 1 to 3 for 'How far do you live from the office?' where 1 represents 'live in Bogotá,' 2 represents 'live near Bogotá,' and 3 represents 'live far from Bogotá (remote work).'

The data was meticulously transformed from an open, unstructured format into a more concise, machine-understandable format. This simplifies data interpretation for developers and minimizes the risk of machine learning errors, ultimately improving accuracy and success rates.

This transformation phase served as a precursor to the next stage of the study following the KDD methodology: feature selection. Two key steps were taken to identify the most significant variables: data cleaning and debugging, followed by prioritizing and renaming the crucial data as determined by expert interviewers from both companies.

The reduction process focused on eliminating redundant questions. For example, if an interviewee was asked, 'Are you familiar with creating test cases?' (yes/no) and "How knowledgeable are you about test case creation?" (1 to 5), the answer to the second question was considered, as it inherently covers the first question.

In order to enhance the predictive capability of the evaluation system, a comprehensive review was carried out to eliminate redundant or irrelevant questions. As a result, a set of nine essential characteristics was identified to ensure the effectiveness of predicting candidates' performance while avoiding redundancy.

Questions such as 'What is your knowledge of programming languages?' were omitted as they did not contribute to evaluating a QA analyst's performance. This refined approach ensures the efficiency and accuracy of the selection process.

Furthermore, to improve readability and clarity, the categorical data in each dataset were renamed. This process

involved transforming complex questions into concise titles with numerical values under a standard scale, simplifying the information and making it more accessible.

A. Source Code Parameters

Specialized machine learning libraries were used to implement and evaluate the algorithms efficiently. These libraries became the technological basis for building and evaluating the models. Some key libraries that played a fundamental role are pandas, scikit-learn, matplotlib.pyplot, and numpy.

Consistency and reproducibility of results are essential in scientific research. To ensure these qualities, specific versions of the following libraries were used.

- pandas version: 2.0.3
- scikit-learn version: 1.3.0
- matplotlib version: 3.7.2
- numpy version: 1.25.0

V. Results

This section presents the meticulous results of a comprehensive comparative evaluation of nine prominent machine learning algorithms for a classification problem. The evaluation was meticulously focused on basic performance metrics, computational efficiency, and discriminative ability. It culminated in adapting Gartner's magic table to provide a thorough analysis. Each of the algorithms, including Nearest Neighbors, Linear SVM, Radial SVM, Gaussian Process, Decision Tree, Neural Network, Logistic Regression, Naive Bayes, and QDA, underwent a meticulous analysis to assess their suitability for application in QA recruitment.

A. Evaluation Metrics

The results will be thoroughly examined, including performance metrics such as precision, accuracy, F1-score, recall, and discriminative ability measured by the area under the Receiver Operating Characteristic (ROC) curve. These metrics are crucial for understanding the algorithm's capabilities. Additionally, the computational efficiency of each algorithm will be analyzed, considering training time and memory usage. These metrics will be described below, including their mathematical expressions and interpretations.

1) Accuracy Rating Score: An analysis of various machine learning algorithms for predicting employee hiring shows an average accuracy of 0.789. The Decision Tree algorithm demonstrates the highest accuracy at 0.8833, while the Linear SVM and QDA exhibit the lowest accuracy at 0.7100. The high accuracy of the Decision Tree can be attributed to its capability to handle complex decision boundaries and nonlinear relationships between features. In contrast, the Linear SVM and QDA, which assume linear relationships and Gaussian distributions, respectively, do not capture the complexities of the data well, explaining their lower performance (see Table I).

2) Precision and Recall Ratios: The overall precision score for all algorithms is 0.764, indicating the algorithms' ability to avoid false positives. With an average recall score of 0.864, the algorithms also demonstrate their proficiency in identifying true positives. The Decision Tree algorithm stands out with the highest precision 0.8833 and recall 0.9021, demonstrating its superior predictive abilities. Conversely, the Linear SVM exhibits the lowest precision 0.6443, while the Gaussian Process displays the lowest recall 0.8322, indicating challenges in identifying true positives. These scores provide valuable insights into the algorithms' predictive strengths and weaknesses.

The superior performance of the Decision Tree can be attributed to its capability to handle complex decision boundaries, unlike the Linear SVM, which relies on a linear hyperplane and may struggle to capture the complexity of the data. The intricate and nonlinear nature of the dataset favors algorithms like Decision Trees, which do not assume a specific data distribution. In contrast, algorithms such as Linear SVM and QDA, which are based on linear assumptions or specific distributions, show lower accuracy and recall, as shown in Table II.

3) F1-Score: The average F1-score across the algorithms is 0.809. The Decision Tree stands out as the most effective, achieving an F1-score of 0.8805, while the Linear SVM lags with a score of 0.7418. The F1-score, which strikes a balance between precision and recall, serves as a measure of a model's effectiveness by accounting for false positives and negatives. The Decision Tree's high F1-score can be attributed to its capability to model nonlinear relationships without making prior assumptions about data distribution, making it particularly valuable when dealing with nonlinear relationships or datasets containing numerous categorical variables.

On the other hand, the low scores of the Linear SVM suggest challenges in handling complex and diverse data. Although Linear SVMs perform well in high-dimensional spaces, they struggle to capture nonlinear relationships without utilizing kernel tricks. (See Table III for details).

The differences in F1-scores underscore the importance of choosing a suitable model for the data set. Although the Decision Tree is practical, each algorithm's performance varies according to the dataset's characteristics, preprocessing, and hyperparameters.

4) Area Under ROC Curve: The average AUC-ROC score among the evaluated algorithms is 0.806, highlighting its ability to differentiate between hires and non-hires. The Decision Tree is the most effective, with an AUC-ROC of 0.8842, indicating its ability to discriminate between classes. In contrast, the Linear SVM has the lowest AUC-ROC of 0.7173, suggesting difficulty separating classes. The Decision Tree's success is due to its ability to handle complex and nonlinear relationships between variables.

The Linear SVM's low AUC-ROC score can be attributed to its focus on finding a linear hyperplane to separate classes. Although effective on data sets with clear linear separation, this approach may fail in more complex scenarios, reducing its ability to discriminate between classes effectively. Refer to Table IV for further details.

B. Learning Curve

This analysis examines the performance of several machine learning algorithms using learning curves, providing a comprehensive evaluation of their behavior against different amounts of training and test data. This approach allows identifying the algorithms' generalization capability and tendency to overfit.

TABLE I
RESULTS OF DIFFERENT MACHINE LEARNING ALGORITHMS

Algorithm	Nearest Neighbors	Linear SVM	Radial SVM	Gaussian Process	Decision Tree	Neural Network	Logistic Regression	Naive Bayes	QDA
Accuracy	0.7733	0.7100	0.8167	0.8267	0.8833	0.8433	0.7633	0.7733	0.7100

TABLE II
PRECISION AND RECALL RATIOS CALCULATED BY CLASSIFIER ALGORITHM

Algorithm	Nearest Neighbors	Linear SVM	Radial SVM	Gaussian Process	Decision Tree	Neural Network	Logistic Regression	Naive Bayes	QDA
Precision	0.7095	0.6443	0.7785	0.8095	0.8600	0.8200	0.7143	0.7110	0.8255
Recall	0.8881	0.8741	0.8601	0.8322	0.9021	0.8601	0.8392	0.8601	0.8601

TABLE III
F1-SCORE CALCULATED BY CLASSIFIER ALGORITHM

Algorithm	Nearest Neighbors	Linear SVM	Radial SVM	Gaussian Process	Decision Tree	Neural Network	Logistic Regression	Naive Bayes	QDA
F1-score	0.7888	0.7418	0.8173	0.8207	0.8805	0.8396	0.7717	0.7785	0.8425

TABLE IV
AUC-ROC CALCULATED BY CLASSIFIER ALGORITHM

Algorithm	Nearest Neighbors	Linear SVM	Radial SVM	Gaussian Process	Decision Tree	Neural Network	Logistic Regression	Naive Bayes	QDA
AUC-ROC	0.7785	0.7173	0.8186	0.8269	0.8842	0.8441	0.7667	0.7708	0.8473

Figure 1 (a) shows that the observed characteristics of Nearest Neighbors are remarkable: it shows rapid initial growth in accuracy, suggesting efficiency with small samples due to its instance-based nature. Both training and test curves stabilize quickly, indicating an efficient balance between bias and variance. In addition, the confidence intervals are narrow, implying low variability in predictions and high certainty in results.

Figure 1 (b) shows that the features observed in the Linear SVM model show moderate accuracy, around 48%, suggesting limitations in capturing the complexity of nonlinear data due to its linear nature. Furthermore, the similarity between the training and test curves indicates that the model is not overfitted but does not improve significantly with increased training data.

In Figure 2 (a), this algorithm shows perfect 100% accuracy on the training set, indicating a complete overfit to the training data. However, in the test set, the high accuracy close to 100% suggests excellent generalization ability despite the perfect fit in training. In addition, the narrow confidence intervals indicate high prediction certainty, suggesting low variability in model performance.

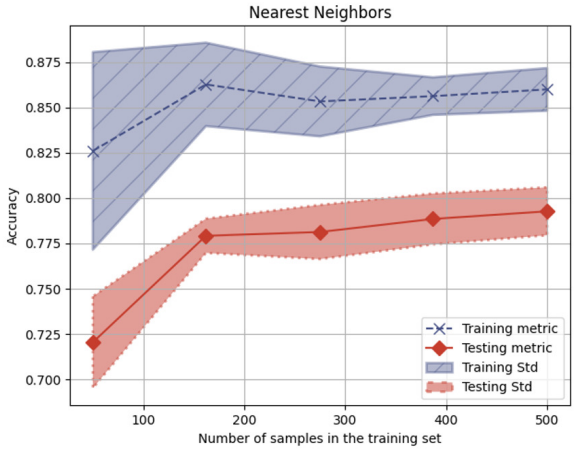
Figure 2 (b) shows that the model effectively captures complex distributions and provides robust predictions. However, it requires substantial computational resources, especially for large volumes of data due to the computation of large matrices. Observed characteristics include an initial variability in accuracy followed by a steady increase, suggesting a period of adjustment before stabilizing and generalizing better with more data. In addition, the continuous improvement in the accuracy of the test set indicates that the model continues to learn and improve its generalization ability. Consistently narrow confidence intervals reflect high prediction certainty and low variability.

In Figure 3 (a), this type of model easily overfits the training data, which limits its generalizability. However, its performance can be improved considerably through techniques such as pruning and ensemble methods such as Random Forests or Boosting. It is especially useful for interpretable problems and with noisy data.

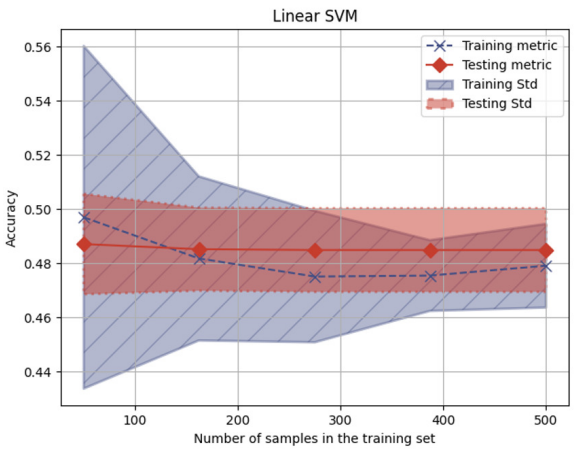
The observed features show constant stability in the training set, suggesting possible overfitting, where the model memorized the training data rather than learning generalizable patterns. In addition, the lack of significant improvement in the accuracy of the test set suggests that the model does not generalize well to new data and may be memorizing the training data.

In Figure 3 (b), it can be seen that Neural Network is suitable for complex problems with large data volumes and nonlinear patterns. They can benefit from using more data and advanced computational resources such as GPUs and regularization techniques to improve generalization and avoid overfitting. The observed features show high initial accuracy, indicating the ability to learn complex patterns in the training data quickly. However, the plateauing in accuracy suggests the need to implement regularization techniques such as Dropout or L2 to improve generalization and avoid overfitting. The moderate width of the confidence intervals reflects reasonable certainty in the predictions, albeit with some variability.

Figure 4 (a) shows that Logistic Regression suits simple linear problems requiring a clear model interpretation. The higher uncertainty in the predictions can be mitigated with more data or regularization techniques. The observed characteristics show wide confidence intervals, indicating higher uncertainty in the model predictions. In addition, the intermediate and stable accuracy suggests acceptable but not optimal model performance.

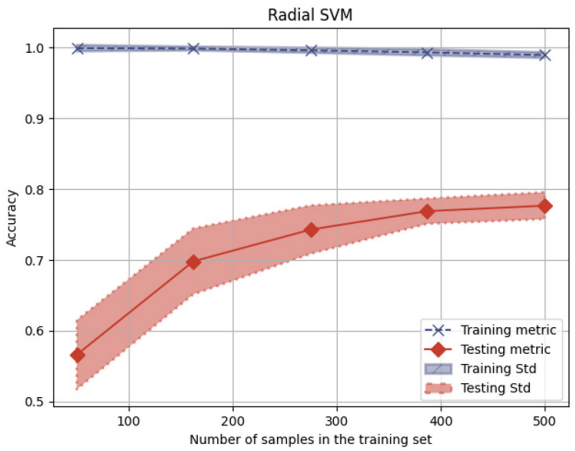


(a) Algorithm learning curve as a function of cross-validation Nearest Neighbors.

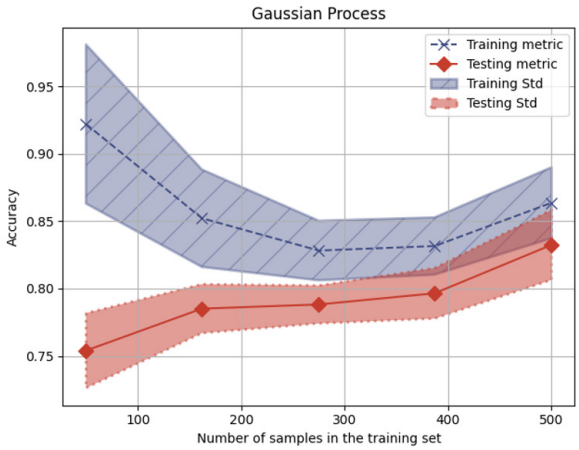


(b) Algorithm learning curve as a function of cross-validation Linear SVM.

Fig. 1. Algorithm learning curve as a function of cross-validation for Nearest Neighbors and Linear SVM



(a) Algorithm learning curve as a function of cross-validation Radial SVM.



(b) Algorithm learning curve as a function of cross-validation Gaussian Process.

Fig. 2. Algorithm learning curve as a function of cross-validation for Radial SVM and Gaussian Process

As shown in Figure 4 (b), Naive Bayes is efficient and robust for classification problems where features are independent. However, it is not suitable for capturing complex interactions between features. The observed features show high stability in accuracy due to the model's conditional independence assumptions, which makes it robust and efficient. In addition, the narrowness of the confidence intervals reflects low variability in the predictions and high certainty.

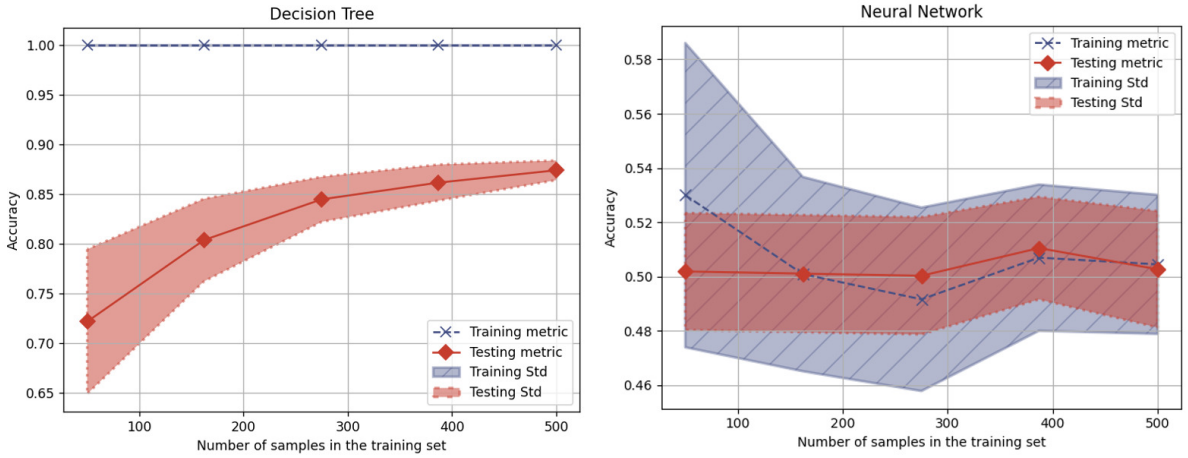
Figure 5 demonstrates that his algorithm is useful for problems with complex nonlinear relationships but can be prone to overfitting in high-dimensional data sets. Regularization or dimensionality reduction techniques are necessary to improve stability and control prediction variability. The observed features show high accuracy variability, reflecting the model's ability to capture complex quadratic relationships in the data. In addition, the improvement in the accuracy of the test set with more data indicates good generalization ability, albeit with higher uncertainty due to the complexity of the model.

Analyzing the learning curves of various machine learning algorithms provides critical insights for selecting the most suitable model. The selection of the correct algorithm for QA applications depends on multiple critical factors:

- Nearest Neighbors and Gaussian Process are recommended for problems where high initial accuracy is required and moderate data size is available.
- Radial SVM is powerful for problems with nonlinear decision boundary but requires careful parameter tuning to avoid overfitting.
- Decision Tree and Neural Network are effective for complex problems, although they require pruning and regularization techniques to mitigate overfitting.
- Logistic Regression and Naive Bayes are efficient for linear and straightforward classification problems, providing interpretability and robustness.
- QDA is suitable for data with complex nonlinear relationships but requires careful dimensionality management and regularization techniques to control variability and improve prediction certainty.

C. Decision Boundaries Map

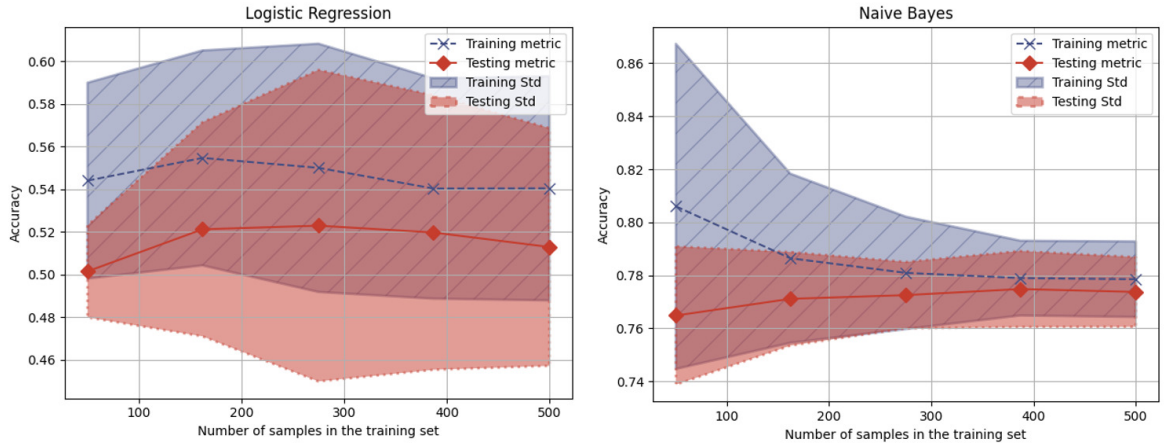
This example illustrates the nature of the decision boundaries of different classifiers. In higher-dimensional spaces, it becomes easier to linearly separate data, and the straightforward nature of classifiers such as Naive Bayes and Linear SVM can lead to improved generalization over other types



(a) Algorithm learning curve as a function of cross-validation Decision Tree.

(b) Algorithm learning curve as a function of cross-validation Neural Network.

Fig. 3. Algorithm learning curve as a function of cross-validation for Decision Tree and Neural Network



(a) Algorithm learning curve as a function of cross-validation Logistic Regression.

(b) Algorithm learning curve as a function of cross-validation Naive Bayes.

Fig. 4. Algorithm learning curve as a function of cross-validation for Logistic Regression and Naive Bayes

of classifiers. Decision boundary mapping is a powerful tool in machine learning, especially for understanding how algorithms classify bivariate data. By mapping the decision boundaries they use to differentiate between classes, this visualizer provides intuitive insight into the behavior and effectiveness of various models.

1) Plotting Quantitative Variables: Specifically, this map plots each data point in a two-dimensional space, defined by two characteristics: the decision to hire a candidate or not. The algorithm then overlays decision boundaries determined by the model to classify the data points into different categories. The areas bounded by these boundaries represent the model's prediction for the class label of the data points within that area. The map shows the training points in solid colors and the test points semi-transparent.

Figure 6 represents the decision map for two quantitative variables in the data set: age versus number of certificates. Similarly, Figure 7 corresponds to age versus number of work experiences. As demonstrated by the Nearest Neighbors algorithm, it displays its decision-making through colored clusters with boundaries that may appear irregular, reflecting its reliance on the proximity of neighboring points to dictate class assignments. This contrasts markedly with the Linear SVM, which opts for a more straightforward approach, manifesting as straight-colored stripes signifying a linear hyperplane designed to separate classes optimally.

The Radial SVM and the Gaussian Process classifier introduce complexity through curved decision boundaries, capturing nuanced patterns within the data. On the other hand, the Decision Tree classifier presents rectangular segments, reflecting its hierarchical decision-making process. Neural Network can generate simple or intricate decision boundaries, adapting to various patterns. Logistic Regression and Naive Bayes produce linear or slightly curved boundaries, while QDA allows more flexible decision boundaries with quadratic functions to capture nonlinear relationships more accurately.

2) Plot Categorical Variables: When transitioning from continuous to categorical variables (as shown in Figure 8), the representation and interpretation of decision boundaries experience a significant shift. Instead of lines or curves demarcating the space, categorical variables produce a visual divided into distinct blocks or regions, each corresponding to a specific combination of category states. This creates

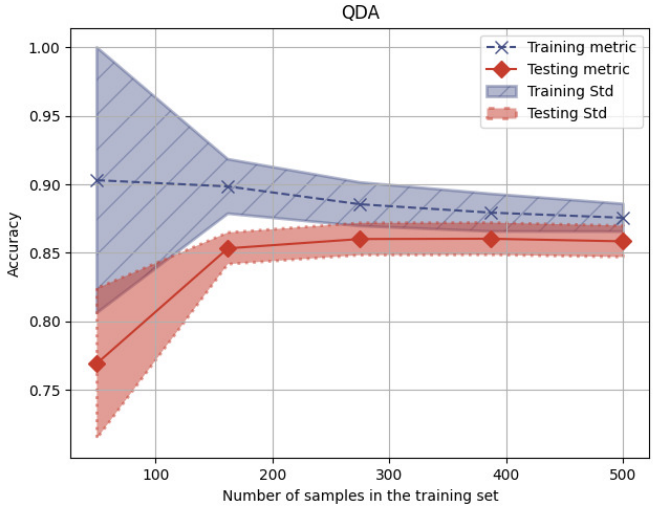


Fig. 5. Algorithm learning curve as a function of cross-validation for QDA

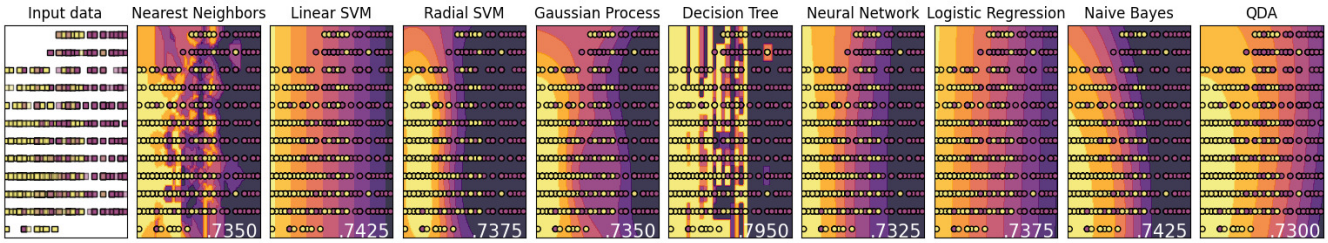


Fig. 6. Decision boundaries map for age vs number of certificates

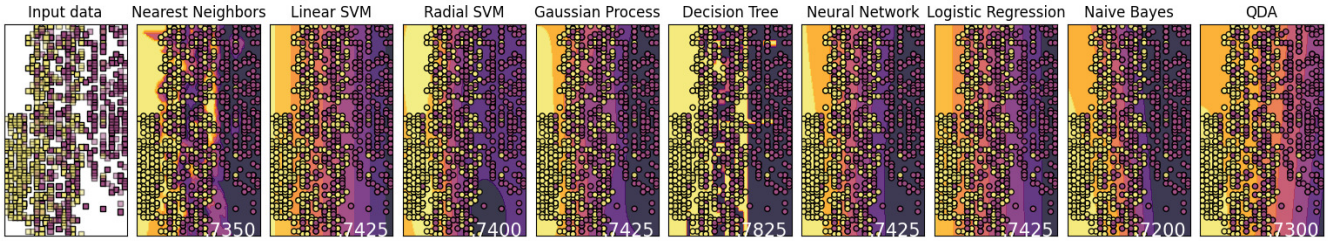


Fig. 7. Decision boundaries map for age vs lenght of work experience

a mosaic-like appearance, with each block representing the predicted class for that particular combination of categorical inputs.

For algorithms like Nearest Neighbors, Linear SVM, and Radial SVM, continuous decision lines or curves are replaced by segmented areas, reflecting a more discrete classification approach. Decision Tree and Neural Network naturally adjust to this change, illustrating their decision-making through clear segmented blocks corresponding to categorical combinations. In this categorical context, Logistic Regression, Naive Bayes, and QDA map the decision space block-like manner, indicating the predicted class for each combination of categories. The overall result is a map showcasing the classification strategy of each algorithm in a discrete categorical landscape, providing a different perspective on how different models handle categorical data.

3) Regularization: Control Overfitting: Regularization is a technique to prevent overfitting by adding a penalty to the size of the coefficients in the learning model, thus discouraging overly complex models. In a Multi-Layer Perceptron (MLP), the alpha parameter acts as a regularization term, controlling the complexity of the model by fitting. Figure 9 compares different alpha values in a data set, showing how they affect the decision functions of the MLP. The lowest alpha values produce the best results when creating classification clusters.

D. Confusion Matrix

This section explores confusion matrices, which are crucial for evaluating the performance of the nine machine learning algorithms studied in candidate classification in a quality control environment. This analysis is essential to determine the accuracy and efficiency of each algorithm, which is essential for selecting the most suitable algorithm for specific QC applications. The matrices are based on a classification scenario for hiring, where the positive and negative results represent the prediction of hiring or not hiring a candidate.

Tables V and VI show the confusion matrices of the Nearest Neighbors and Linear SVM algorithms, evaluating their performance. Both reveal a worrying trend with a significant

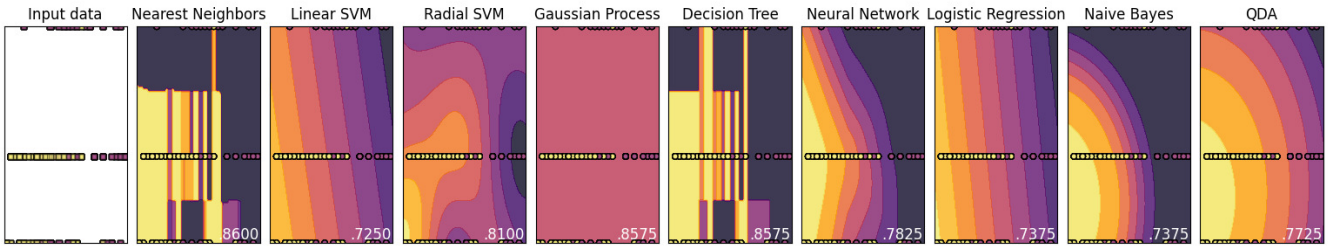


Fig. 8. Decision boundaries map for age vs length of salary aspiration

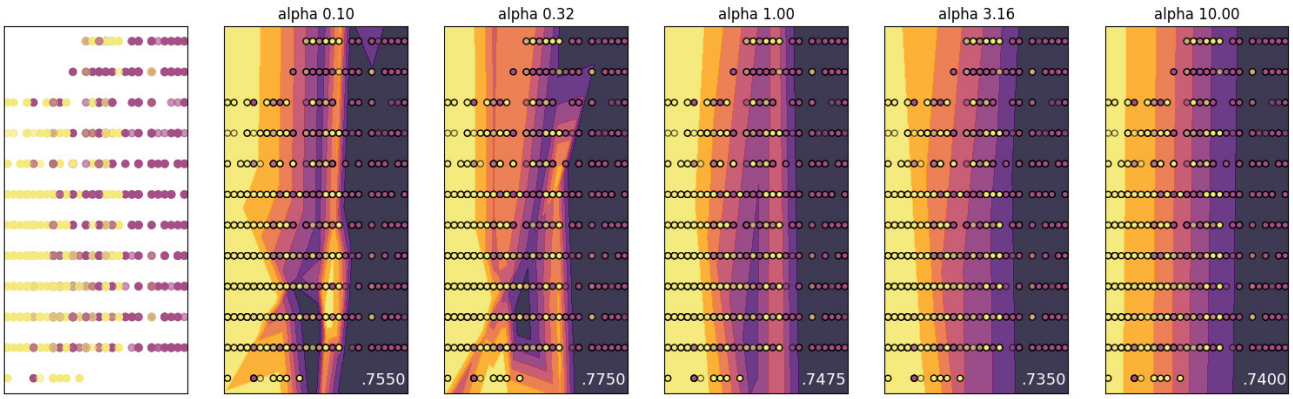


Fig. 9. Decision boundaries map for age vs number of certificates with alpha parameter variation

TABLE V
NEAREST NEIGHBORS CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	127	16
Actual negative	52	105

TABLE VI
LINEAR SVM CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	125	18
Actual negative	69	88

presence of False Positives (FP), raising doubts about their suitability for quality control contracting processes. On the other hand, the Radial SVM in Table VII stands out as a model with minimal errors and a high hit rate of 81%, outperforming other algorithms and being a solid option for quality selection support.

Tables VIII, IX, and X show the results of the Gaussian Process, Decision Tree, and Neural Network models, revealing that they are competent in accurately predicting positive and negative classifications. They stand out for their high hit rates, with Decision Tree leading with 88.3%, followed by Gaussian Process with 82.6% and Neural Network with 84.3%. Their solid performance positions them as promising options for support in quality selection processes.

TABLE XI
LOGISTIC REGRESSION CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	120	23
Actual negative	48	109

TABLE VII
RADIAL SVM CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	123	20
Actual negative	35	122

TABLE VIII
GAUSSIAN PROCESS CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	119	24
Actual negative	28	129

TABLE XII
NAÏVE BAYES CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	123	20
Actual negative	50	107

TABLE XIII
QDA CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	123	20
Actual negative	26	131

Tables XI, XII, and XIII detail the performance of the Logistic Regression, Naive Bayes, and QDA algorithms. Logistic Regression and Naive Bayes showed many false positives (FP), negatively affecting their QA selection process reliability. Although QDA had a similar false negative (FN) value, it had half as many FPs as the other two algorithms. The hit rates were 76.3%, 76.6%, and 84.6% for Logistic Regression, Naive Bayes, and QDA, respectively, highlighting the latter as a strong candidate to support QA

TABLE IX
DECISION TREE CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	129	14
Actual negative	21	136

TABLE X
NEURAL NETWORK CONFUSION MATRIX

	Predicted positive	Predicted negative
Actual positive	123	20
Actual negative	27	130

selection processes.

E. Matthews Correlation Coefficient (MCC)

Classifiers' performance on a classification problem is essential for its practical application. The MCC, which considers true positives, true negatives, false positives, and false negatives, is a robust measure to evaluate their effectiveness, especially in situations of class imbalance or asymmetric costs associated with classification errors.

Figure 11 (a) presents the MCC values for each classifier evaluated. Highlighting the Decision Tree obtains an MCC of 0.7675, demonstrating its robustness in class prediction. This result is attributed to its ability to capture nonlinear relationships and handle complex data sets. On the other hand, Gaussian Process 0.6532, Neural Network 0.6874, and QDA 0.6938 also show competitive performance, exceeding an MCC of 0.65, indicating their ability to model complex relationships.

In contrast, Nearest Neighbors 0.5670, Linear SVM 0.4541, Logistic Regression 0.5367, and Naive Bayes 0.5476 exhibit inferior performance. Although they may be more computationally efficient or interpretable, their ability to capture nonlinear relationships or model complex data may be limited in this specific data set.

1) Analysis of the Balance between Bias and Variance in Classification Algorithms: Balancing bias and variance is crucial in machine learning models, as it influences their generalizability. Looking at Figure 11 (b), the relationship for each algorithm evaluated can be analyzed.

- Decision Tree: Shows an accurate fit in the training set but overfitting in the test set, indicating high variance.
- Neural Network: This has a similar pattern, showing that it overfits the training data.
- Radial SVM and Gaussian Process: They have an improved balance between bias and variance, offering better generalization.
- Naive Bayes and Logistic Regression: They present moderate bias and low variance, providing stability in both data sets.-suited for applications where interpretability and stability are crucial.

Figure 11 (b) illustrates how the balance between bias and variance influences each algorithm's predictive performance and generalization.

F. Complexity Analysis

The balance between training time and cross-validation score can now be evaluated. In the following figures, it is essential to determine when the cross-validation score stops increasing while the training time continues to grow.

The Nearest Neighbors algorithm in Figure 12 (a) shows a strong correlation between fitting time and accuracy, reaching its maximum around 0.80 with moderate fitting times, between 0.001 and 0.0026 seconds. However, the Linear SVM in Figure 12 (b) maintains a constant accuracy of around 0.49 regardless of the fitting time, which varies up to 14 seconds. This stability suggests that allocating more computational resources does not significantly improve predictive performance beyond a certain point, indicating a possible inefficiency for this dataset.

The Radial SVM in Figure 13 (a) shows a significant improvement in accuracy with increasing fitting time, reaching approximately 0.8 with 0.08 seconds of fitting. Although the confidence intervals narrow with longer times, they indicate variability in performance, although the overall predictive ability remains robust. On the other hand, the Gaussian Process model in Figure 13 (b) achieves the highest accuracy, up to 0.9, but with an extended fitting time of up to 14 seconds, reflecting greater computational demands. The wide confidence intervals suggest variability in performance, possibly due to model sensitivity to hyperparameters and data characteristics.

The Decision Tree algorithm in Figure 14 (a) shows a noticeable increase in accuracy as the fitting time increases, reaching a maximum near 0.88. Fit times remain relatively short, with a maximum of 0.004 seconds, and the narrowing confidence intervals indicate consistent performance with adequate training. In contrast, the Neural Network model in Figure 14 (b) exhibits fluctuating accuracy within a fitting time interval of 0.05 to 0.085 seconds. Although it achieves a maximum accuracy of about 0.54, the wide confidence intervals suggest significant variability, possibly due to the model's sensitivity to initialization and training parameters.

In Figure 15 (a), the Logistic Regression model exhibits a constant accuracy of about 0.52 at different fitting times, concentrated around 0.004 seconds. The narrow confidence intervals indicate stable performance, albeit with moderate accuracy, reflecting the simplicity and reliability of the model. Meanwhile, in Figure 15 (b), the Naive Bayes model shows a gradual increase in accuracy as the fitting times increase, reaching approximately 0.78 accuracy with narrow confidence intervals, indicating robust performance with minimal computational requirements.

In Figure 16, the QDA model shows consistent levels of accuracy, reaching approximately 0.86, with adjustment times mainly around 0.0016 seconds, highlighting its efficiency.

Gaussian Process and Decision Tree models are highly accurate, although the former requires longer fitting times. On the other hand, Linear SVM and Neural Network have relatively lower accuracy. QDA and Naive Bayes offer efficient training times with moderate accuracy, balancing accuracy, and computational efficiency.

G. Gartner Magic Quadrant (GMQ)

A customized GMQ will be implemented to obtain a comprehensive and detailed view of the nine algorithms under consideration. This approach will facilitate a systematic and

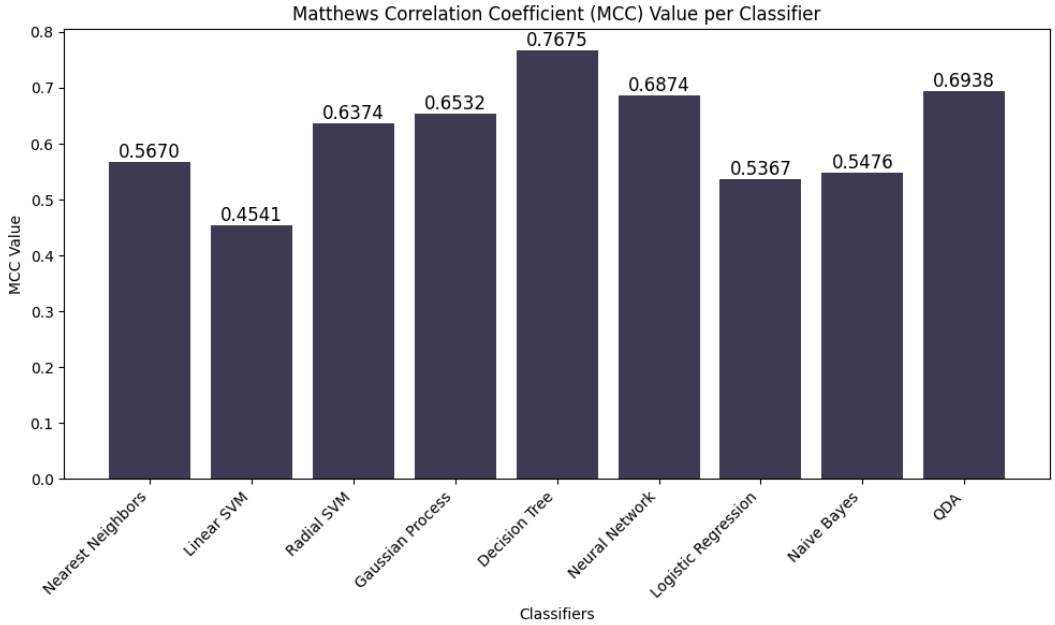


Fig. 10. MCC value per classifier and training set vs testing set

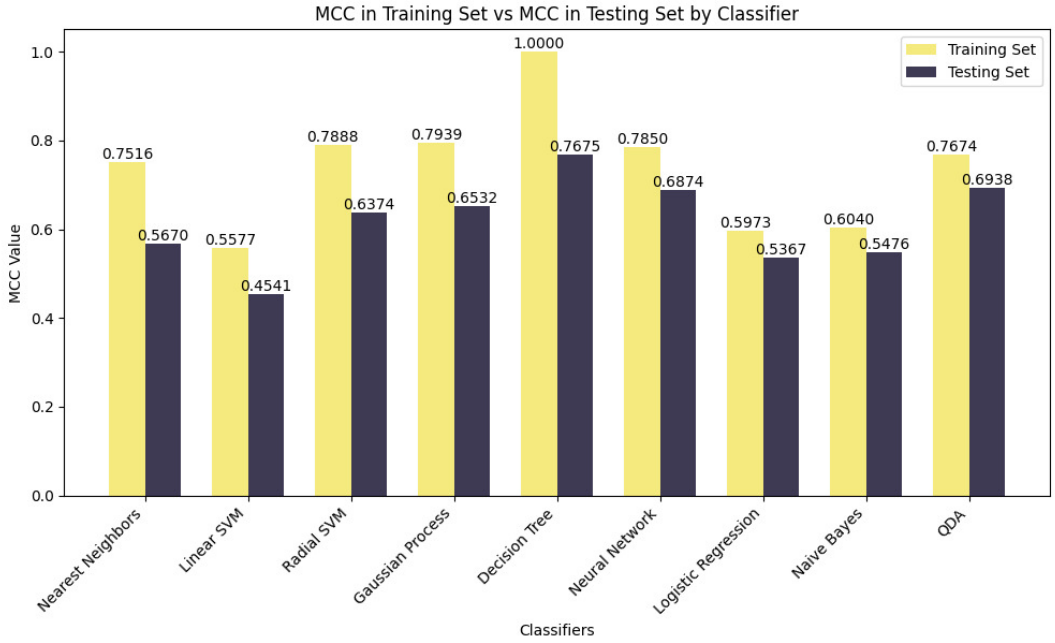


Fig. 11. MCC value per classifier and training set vs testing set

thorough evaluation and comparison of each algorithm. The Magic Quadrant will be specifically tailored for these nine algorithms, providing a clear visual representation of their performance relative to the metrics analyzed in this paper.

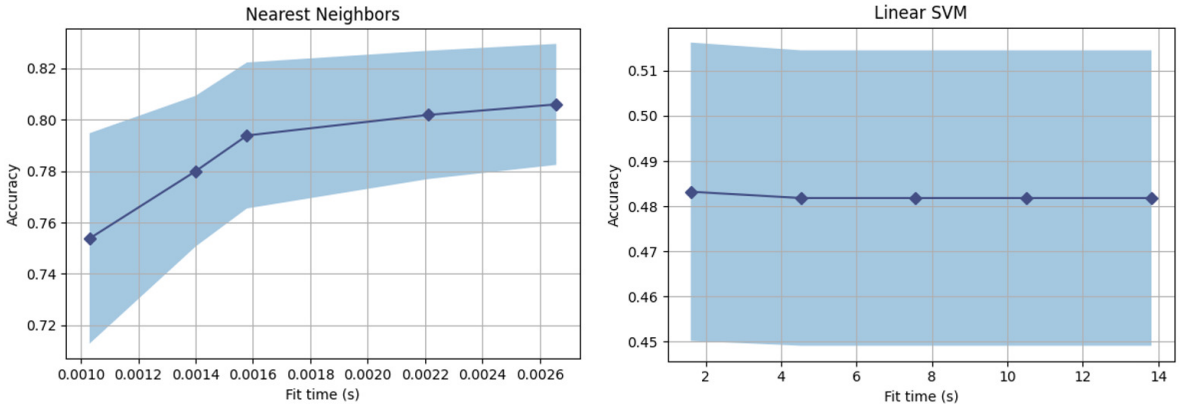
Each algorithm will be positioned in the chart according to its performance on critical dimensions assessed in this study. By evaluating these factors, it will be possible to identify each algorithm's strengths and areas for improvement and how they compare to each other in a practical context.

This methodology will objectively determine the most suitable algorithm for the study's specific needs. Through this evaluation, an informed selection can be made that maximizes the benefits and minimizes the constraints, thus ensuring the optimal algorithmic solution is chosen for the case study.

In the Cartesian plane representing the Gartner Magic Box, the x-axis signifies the algorithms' performance across various metrics, including Accuracy (Acc in Equation (1)), Precision (Pre in Equation (1)), recall, F1-score, AUC-ROC, and MCC. Each of these metrics holds equal significance along the x-axis for every algorithm. Consequently, the average of these metric values is computed for each algorithm to establish its position on the x-axis of the GMQ. The equation used to calculate the x-axis value is as follows:

$$\text{x-axis} = \frac{\text{Acc} + \text{Prec} + \text{Recall} + \text{F1-score} + \text{AUC-ROC} + \text{MCC}}{6} \tag{1}$$

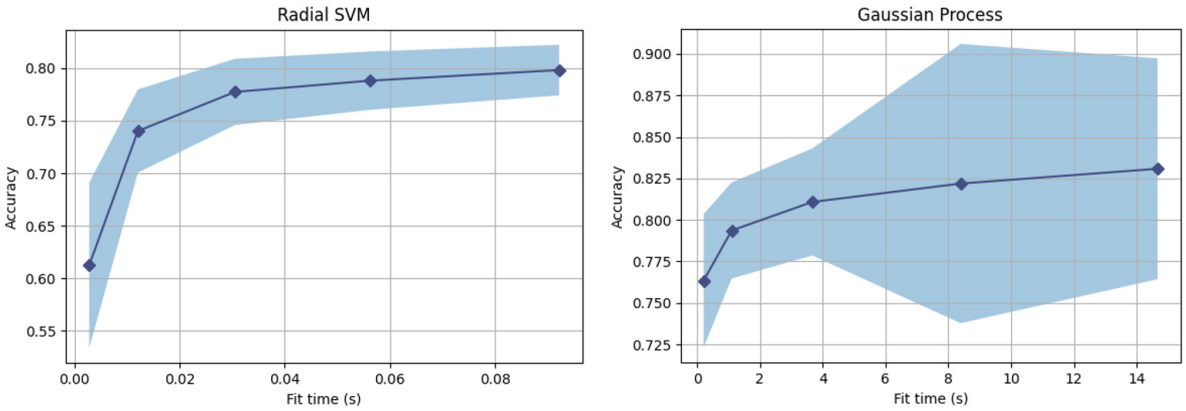
Each metric is given equal weight to determine the algo-



(a) Nearest Neighbors complexity curves.

(b) Linear SVM complexity curves.

Fig. 12. Nearest Neighbors and Linear SVM complexity curves



(a) Radial SVM complexity curves.

(b) Gaussian Process complexity curves.

Fig. 13. Radial SVM and Gaussian Process complexity curves

rithm's positioning on the GMQ's x-axis.

The y-axis of the GMQ, known as the Comprehensive Evaluation Axis, utilizes a wider range of evaluation criteria. These criteria are based on a combination of learning curve, decision boundaries maps, confusion matrices, and complexity analysis. Each criterion provides an objective evaluation value. The method for obtaining these values depends on the specific characteristics of each of the three evaluation criteria. The values considered for each criterion are detailed below:

1) Learning Curve: The learning curves illustrate how the model's performance improves with more data. In this instance, the stability and convergence speed of the model will be assessed.

Stability: This is determined based on the standard deviation of the points on the learning curve. A lower standard deviation indicates greater stability, and it will be calculated using Equation (2).

$$\text{Stability} = 1 - \left(\frac{\sigma_{\text{learning curve}}}{\sigma_{\max}} \right) \tag{2}$$

Convergence: This is measured by the amount of data required to reach a specific percentage of the maximum yield (e.g., 95%). This will be calculated using Equation (3).

$$\text{Convergence} = 1 - \left(\frac{N_{\text{samples at 95\%}}}{N_{\max}} \right) \tag{3}$$

An average of the stability and convergence values will be computed for each algorithm to determine the learning curve criterion. As per Equation (4), both stability and convergence are equally important for all algorithms.

$$\text{Learning Curve} = \frac{\text{Stability} + \text{Convergence}}{2} \tag{4}$$

2) Decision Boundaries Maps: In this context, two specific metrics will be assessed in relation to the three maps generated in the study:

Clarity: This metric quantifies the variation within decision regions, with lower variation indicating greater clarity.

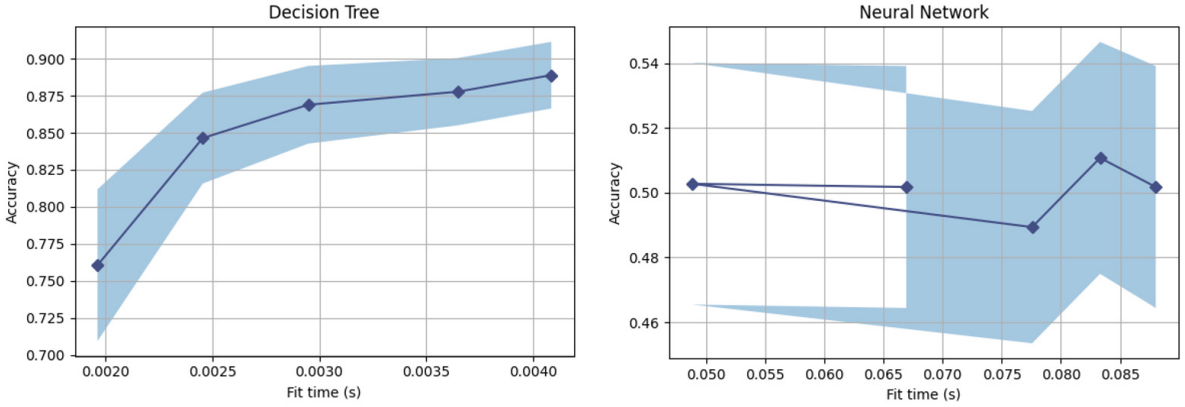
$$\text{Clarity} = 1 - \left(\frac{\text{Boundary Variation}}{\text{Boundary Variation}_{\max}} \right) \tag{5}$$

Accuracy: It measures the proportion of samples correctly classified within the map.

$$\text{Accuracy} = \frac{N_{\text{correctly classified}}}{N_{\text{total}}} \tag{6}$$

To derive the criterion value of the decision boundaries maps, the following steps should be followed: For each of the three generated maps, the average of the clarity and precision values for each algorithm will be calculated.

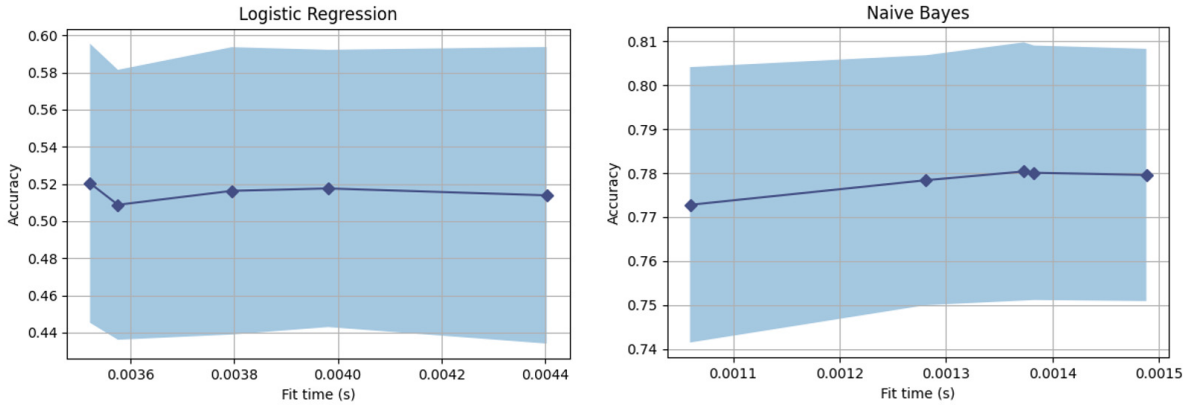
$$\text{Clarity}_T = \frac{\text{Clarity}_1 + \text{Clarity}_2 + \text{Clarity}_3}{3} \tag{7}$$



(a) Decision Tree complexity curves.

(b) Neural Network complexity curves.

Fig. 14. Decision Tree and Neural Network complexity curves



(a) Logistic Regression complexity curves.

(b) Naïve Bayes complexity curves.

Fig. 15. Logistic Regression and Naive Bayes complexity curves

$$\text{Accuracy}_T = \frac{\text{Accuracy}_1 + \text{Accuracy}_2 + \text{Accuracy}_3}{3} \quad (8)$$

Subsequently, the average of the three values obtained for each algorithm will be calculated, as all comparisons are considered equally important.

$$\text{Decision Boundary} = \frac{\text{Clarity}_T + \text{Accuracy}_T}{2} \quad (9)$$

Equations (7), (8), and (9) provide the mathematical foundation for these calculations, ensuring a comprehensive and equitable evaluation of the algorithms' performance in terms of the clarity and accuracy of the decision boundaries.

3) Confusion Matrix: The confusion matrices assesses a model's performance based on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Two specific metrics are used to evaluate the model: clarity and balance of classifications.

Clarity: This is determined by the ratio of the sum of true positives and true negatives to the total number of samples. The formula for this metric is given in Equation (10):

$$\text{Clarity} = \frac{TP + TN}{N_{\text{total}}} \quad (10)$$

Balance: This is calculated as the average of precision and recall. The formula for this metric is given in Equation (11):

$$\text{Balance} = \frac{\text{Precision} + \text{Recall}}{2} \quad (11)$$

To derive the value of the confusion matrix criterion, an average of the clarity and balance values will be computed for each algorithm, as both metrics are equally important across all algorithms. See Equation (12) for the combined representation of these metrics.

$$\text{Confusion Matrix} = \frac{\text{Balance} + \text{Clarity}}{2} \quad (12)$$

4) Complexity Analysis: Complexity analysis assesses the efficiency and stability of machine learning algorithms during training. To do this, three specific metrics derived from the accuracy curves will be examined as a function of the fitting time:

AUC-ROC - accuracy vs. fit time: The AUC.ROC is computed for the accuracy curve over fit time. This provides a measure of the model's efficiency and performance over time (see Equation (13)).

$$\text{AUC-ROC} = \frac{\text{AUC-ROC of accuracy vs. Fit time}}{\text{AUC-ROC}_{\text{max}}} \quad (13)$$

Convergence time: This is calculated as the time taken for the model to achieve 95% of its maximum accuracy (see Equation (14)).

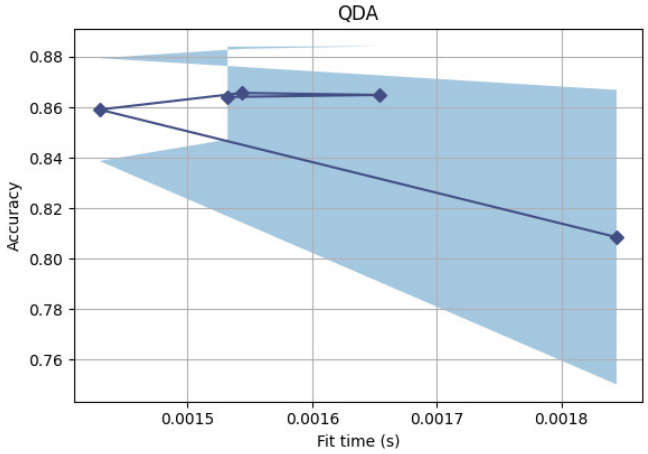


Fig. 16. QDA complexity curves

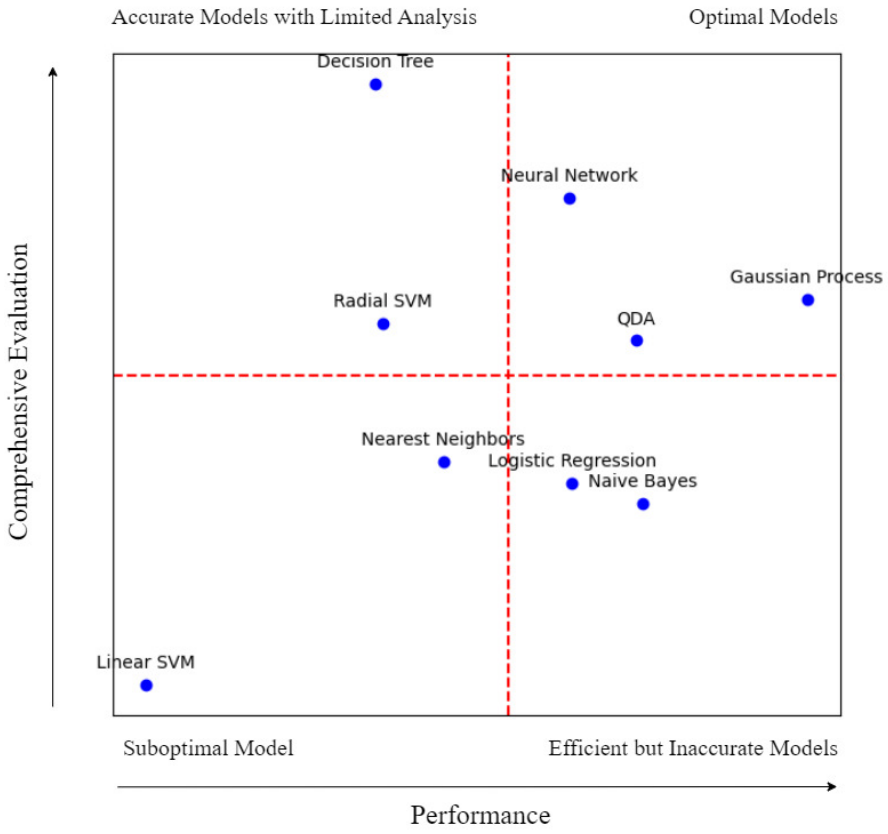


Fig. 17. Algorithm comparison using the GMQ

$$\text{Convergence Time} = 1 - \left(\frac{T_{95\% \text{ max accuracy}}}{T_{\text{max fit time}}} \right) \tag{14}$$

Stability of the curve: Stability is determined as the variability (standard deviation) of the accuracy over time. A lower standard deviation indicates greater stability (see Equation (15)).

$$\text{Stability} = 1 - \left(\frac{\sigma_{\text{accuracy}}}{\sigma_{\text{max}}} \right) \tag{15}$$

To determine the complexity analysis criterion, the average of the AUC-ROC, Convergence Time, and Stability values will be calculated for each algorithm, as both metrics are equally important. Equation (16) presents the corresponding equation for combining these metrics.

$$\text{Complexity} = \frac{\text{AUC-ROC} + \text{Convergence Time} + \text{Stability}}{3} \tag{16}$$

In order to find the value of the y-axis, the average of the values obtained by applying the processes described in equations (4) represented as LDC, (9) represented as DBV, (12) represented as CMV and (16) represented as CV is calculated. This procedure can be expressed by the following equation (17):

$$\text{y-axis} = \frac{\text{LCV} + \text{DBV} + \text{CMV} + \text{CV}}{4} \tag{17}$$

Once the values of both axes have been determined, they are plotted on a Cartesian plane. The distribution of the values results in two intersecting straight lines at the center of the plane, creating four quadrants. In this study, each of these quadrants is named as follows:

- Optimal models: These models demonstrate high performance across all evaluation metrics (accuracy, precision, recall, F1-score, AUC-ROC, MCC) as well as in comprehensive evaluation (learning curve, decision boundaries maps, confusion matrices, complexity analysis). They are ideal for implementation.
- Accurate models with limited analysis: These models exhibit strong evaluation metrics but do not perform as well in performance analysis. They can be chosen when accuracy is crucial but may require optimization to improve efficiency and complexity.
- Suboptimal models: Models in this quadrant do not excel in either evaluation metrics or performance analysis. These models would typically be discarded or considered for significant improvement.
- Efficient but inaccurate models: These models are efficient and yield good results in performance analysis, but are not very accurate according to the evaluation metrics. They could be useful in situations where model efficiency is more important than accuracy.

VI. Conclusions

This research extensively assessed various machine learning algorithms for use in QA Recruiting. This assessment utilized a range of performance metrics (including accuracy, precision, recall, F1-score, AUC-ROC, and MCC) and analysis criteria (such as learning curves, decision boundaries maps, confusion matrices, and complexity analysis). The evaluation enabled the classification of the algorithms into four primary categories: Optimal Models, Accurate Models with Limited Analysis, Suboptimal Models, and Efficient but Inaccurate Models, as illustrated in Figure 17.

Optimal Models: Algorithms in this category, such as Gaussian Process and Neural Network, are notable for their exceptional accuracy and operational efficiency performance. These models are well-suited for complex applications that require effective handling of nonlinear relationships, offering an optimal balance between accuracy and analysis.

Accurate Models with Limited Analysis: Algorithms like Decision Tree demonstrate high accuracy metrics but limited analysis performance. They are suitable for scenarios prioritizing accuracy, although they may require additional optimization to enhance efficiency and complexity management.

Suboptimal Models: Algorithms like Linear SVM do not excel in either accuracy or performance analysis, making them less suitable for most practical applications. These models would require significant improvements before being considered for implementation.

Efficient but Inaccurate Models: Algorithms such as Logistic Regression and Nearest Neighbors are efficient and perform well in performance analysis but do not achieve high levels of accuracy. They are helpful in applications where operational efficiency is more critical than absolute accuracy.

The comparison between the algorithms highlights the importance of choosing the appropriate model based on the dataset's characteristics and the application's specific requirements. This study emphasizes that there is no universally superior model; the choice of algorithm should be based on a trade-off between accuracy, efficiency, and generalizability.

In conclusion, this study's results offer clear and practical guidance for selecting machine learning algorithms, emphasizing the need to consider technical performance and operational efficiency. The ability to choose a suitable model can significantly impact the effectiveness and success of machine learning applications in real-world scenarios, which is the focus of this paper.

Acknowledgment

We sincerely thank the Universidad Distrital Francisco José de Caldas (Bogotá - Colombia) for providing us with the resources and enabling environment to carry out this research.

We thank our families and loved ones for the unwavering support, understanding and encouragement during this process.

References

[1] A. Panesar, "Machine learning and AI for healthcare," Coventry, UK: Apress, vol. -, no. -, pp. 1-73, 2019.

[2] D. Nasrudin, C. Rochman, D. Kuntadi, and D. Jamaluddin, "Research Trends of Quality Assurance in Islamic Education," Journal of Quality Assurance in Islamic Education (JQAIE), vol. 1, no. 1, pp. 22-32, 2021.

[3] N. Sharma, R. Bhutia, V. S. Sardar, A. P. George, and F. S. Ahmed, "Novel Hiring Process using Machine Learning and Natural Language Processing," in 2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1-6, 2021.

[4] K. M. Torres and A. Statti, "Utilizing Technology to Enhance Human Resource Practices," in Designing and Implementing HR Management Systems in Family Businesses, IGI Global, pp. 83-100, 2021.

[5] V. Russo, A. Oriou, F. Huynh, and C. Baron, "Software Quality Assurance Dashboard for Renault Software Robustness plan with SQUORE tool," in 9th European Congress EMBEDDED REAL TIME SOFTWARE AND SYSTEMS (ERTS2), pp. 1-6, 2018.

[6] M. Adnan et al., "Predicting at-risk students at different percentages of course length for early intervention using machine learning models," IEEE Access, vol. 9, pp. 7519-7539, 2021.

[7] F. Macfarlane, J. Duberley, C. Fewtrell, and M. Powell, "Talent management for NHS managers: human resources or resourceful humans?," Public Money & Management, vol. 32, no. 6, pp. 445-452, 2012.

[8] J. Ingebrigtsen, E. Roynstrand, and M. E. Berge, "An evaluation of the preclinical prosthodontic training at the Faculty of Dentistry, University of Bergen, Norway," European Journal of Dental Education, vol. 12, no. 2, pp. 80-84, 2008.

[9] C. Magazzino, M. Mele, and N. Schneider, "Testing the convergence and the divergence in five Asian countries: from a GMM model to a new Machine Learning algorithm," Journal of Economic Studies, vol. 49, no. 6, pp. 1002-1016, 2022.

[10] M. S. Holden et al., "Machine learning methods for automated technical skills assessment with instructional feedback in ultrasound-guided interventions," International Journal of Computer Assisted Radiology and Surgery, vol. 14, pp. 1993-2003, 2019.

[11] B. Mahesh, "Machine learning algorithms-a review," International Journal of Science and Research (IJSR), vol. 9, pp. 381-386, 2020.

[12] T. O. Ayodele, "Types of machine learning algorithms," in New advances in machine learning, vol. 3, pp. 19-48, 2010.

[13] V. Nasteski, "An overview of the supervised machine learning methods," Horizons, vol. 4, pp. 51-62, 2017.

[14] H. Li and M. L. Wong, "Financial fraud detection by using Grammar-based multi-objective genetic programming with ensemble learning," in 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1113-1120, 2015.
[15] Z. Wang, K. Crammer, and S. Vucetic, "Breaking the curse of kernelization: budgeted stochastic gradient descent for large-scale SVM training," J. Mach. Learn. Res., vol. 13, pp. 3103-3131, 2012.
[16] W. Zhang, C. Wu, H. Zhong, Y. Li, and L. Wang, "Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization," Geoscience Frontiers, vol. 12, pp. 469-477, 2021.
[17] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, 2016.
[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.
[19] S. Abou El-Seoud, N. Farag, and G. McKee, "A Review on Non-Supervised Approaches for Cyberbullying Detection," Int. J. Eng. Pedagog., vol. 10, no. 4, pp. 25-34, 2020.
[20] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," 2nd ed., Springer, 2017.
[21] Y. Yang, C. Qian, H. Li, Y. Gao, J. Wu, C. J. Liu, and S. Zhao, "An efficient DBSCAN optimized by arithmetic optimization algorithm with opposition-based learning," The Journal of Supercomputing, vol. 78, no. 18, pp. 19566-19604, 2022.
[22] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems," 2nd ed., O'Reilly Media, 2019.
[23] P.-N. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining," 2nd ed., Pearson, 2019.
[24] J. Han, J. Pei, and M. Kamber, "Data Mining: Concepts and Techniques," 3rd ed., Morgan Kaufmann, 2011.
[25] A. Subero and A. Subero, "Intro to DSA, Types, and Big O," in Codeless Data Structures and Algorithms: Learn DSA Without Writing a Single Line of Code, pp. 3-17, 2020.
[26] F. Taher, N. Prakash, A. Shaffie, A. Soliman, and A. El-Baz, "An Overview of Lung Cancer Classification Algorithms and their Performances," IAENG International Journal of Computer Science, vol. 48, no.4, pp1021-1027, 2021.
[27] Mohammed S. Hashim, and Ali A.Yassin, "Breast Cancer Prediction Using Soft Voting Classifier Based on Machine Learning Models," IAENG International Journal of Computer Science, vol. 50, no.2, pp705-714, 2023.
[28] P. Ou and H. Wang, "Modeling and forecasting stock market volatility by Gaussian processes based on GARCH, EGARCH and GJR models," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2011, WCE 2011, 6-8 July, 2011, London, U.K., pp 338-342.
[29] M. Krzywinski and N. Altman, "Classification and regression trees," Nature Methods, vol. 14, no. 8, pp. 757-758, 2017.
[30] Yuxia Sun, Yanjia Chen, Yuchang Pan, and Lingyu Wu, "Android Malware Family Classification Based on Deep Learning of Code Images," IAENG International Journal of Computer Science, vol. 46, no.4, pp524-533, 2019.
[31] Kalhori S.R.N, Nasehi M, and Zeng X.J, "A logistic regression model to predict high risk patients to fail in tuberculosis treatment course completion," IAENG International Journal of Applied Mathematics, vol. 40, no. 2, pp102-107, 2010.
[32] B. E. V. Comendador, L. W. Rabago, and B. T. Tanguilig, "An educational model based on Knowledge Discovery in Databases (KDD) to predict learner's behavior using classification techniques," in 2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1-6, 2016.
[33] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," AI Magazine, vol. 17, no. 3, pp. 37-54, 1996.
[34] H. Chen, R. H. Chiang, and V. C. Storey, "Business intelligence and analytics: From big data to big impact," MIS Quarterly, pp. 1165-1188, 2012.

Revised Date: August 4, 2024.

List of Changes

- The name of the university in Author Information and Acknowledgment sections has been changed to the proper name in the native language of the university, in accordance with the requirements established by that institution.