# Cancer: Investigating the Impact of the Implementation Platform on Machine Learning Models

Adedayo Seun Olowolayemo, Amina Souag, Konstantinos Sirlantzis

School of Engineering, Technology and Design, Canterbury Christ Church University (CCCU)

Canterbury, UK

email: (a.olowolayemo502, amina.souag, konstantinos.sirlantzis)@canterbury.ac.uk

*Abstract*— **In the context of global cancer prevalence and the imperative need to improve diagnostic efficiency, scientists have turned to machine learning (ML) techniques to expedite diagnosis processes. Although previous research has shown promising results in developing predictive models for faster cancer diagnosis, discrepancies in outcomes have emerged, even when employing the same dataset. This study addresses a critical question: does the choice of development platform for ML models impact their performance in cancer diagnosis? Utilizing the publicly available Wisconsin Diagnostic Breast Cancer (WDBC) dataset from the University of California, Irvine (UCI) to train four ML algorithms on two distinct platforms: Python SciKit-Learn and Knime Analytics. The algorithms' performance was rigorously assessed and compared, with both platforms operating under their default configurations. The findings of this study underscore an impact of platform selection on ML model performance, emphasizing the need for thoughtful consideration when choosing a platform for predictive models' development. Such a decision bears significant implications for model efficacy and, ultimately, patient outcomes in the healthcare industry. The source code (Python and Knime) and data for this study are made fully available through a public GitHub repository.**

***Keywords-Cancer; Machine Learning; Python SciKit-Learn; Knime Analytics; Wisconsin Diagnostic Breast Cancer (WDBC).***

## I. INTRODUCTION

Cancer is a global health menace responsible for nearly 10,000,000 deaths in year 2020 alone [1][2][3]. This disease is characterized by the uncontrolled growth of body cells which forms *tumors* classified as *malignant* - the cancerous cells that are invasive and capable of spreading to other parts of the body - or *benign* - the non-cancerous cells that are not capable of invading nearby tissues and are less harmful. This disease's complexity spans multiple organs like the breast, kidneys, brain, lungs, prostate, ovaries, and skin, posing substantial challenges for healthcare professionals and patients alike. Despite significant progress in cancer understanding and treatment development, timely diagnosis remains critical as delays exacerbate patients' conditions, often leading to irreparable outcomes and increased mortality rates.

Scientists are channeling substantial resources into accelerating the diagnostic process, and artificial intelligence, which has proven effective in various industries, is offering hope for quicker and more effective cancer diagnosis methods. Machine learning, a subset of artificial intelligence, has profoundly reshaped medical research, enhancing diagnostic precision, prognostic accuracy, and treatment strategies. By harnessing advanced computational techniques, ML algorithms ranging from Logistic Regression (LR) to Decision Trees (DT), Random Forests (RF), Gradient Boosting (GB) among several others for cancer diagnosis, extract insights from intricate medical data used in revolutionizing clinical decision-making and improving patient outcomes from pinpointing diseases through image analysis [4] to forecasting patient responses to therapies [5].

These ML algorithms have showcased remarkable potential in the field. However, a critical aspect that we found to be underexplored is the impact of implementation platforms on which the algorithms are trained, and models are developed, such as Python Scikit-learn and Knime analytics, on the performance of these algorithms. Therefore, understanding the nuanced influence of implementation platforms on ML algorithms is pivotal.

Against this backdrop, this study used supervised learning, training models on labeled WDBC datasets [6] to evaluate the performance metrics of ML algorithms including accuracy, precision, recall, and F1-Score focusing on the nuanced relationship between implementation platforms and the efficacy of these algorithms. It emphasizes the potential impact of platform choice on algorithm behavior, highlighting the necessity of discerning these disparities.

This study embarks on two pivotal inquiries:

(1) It seeks to answer whether the choice of the implementation platform significantly impacts the performance of ML algorithms in cancer data classification,

(2) Identifies which of the selected algorithms performed best in cancer dataset binary classification task.

By delving into these fundamental questions and meticulously avoiding hyperparameter tuning, this research provides nuanced insights, offering a comprehensive understanding of the intricate interplay between ML

ISBN: 978-1-68558-136-7

algorithms, implementation platforms, and feature significance.

The rest of this paper is organized as follows: Works relating to this study were explored in Section II, examining relevant literature to the research question. The section starts by looking at studies that used ML in cancer research, then at the different algorithms implemented, the train-test split, performance metrics, dataset sources, and implementation platforms used. Section III outlines the Methodology used for this study, detailing data collection and pre-processing steps, feature selection, and implementation of the selected ML models. Section IV presents the Results and Discussion, followed by Conclusion and Future Work in Section V.

## II. RELATED WORK

Researchers have explored and reported the use of various supervised ML algorithms in different areas of human health and medical fields. Some previous studies reviewed are briefly discussed below.

### A. *Machine Learning in Cancer Research*

Michael et al. in [7] tested five ML classification algorithms on 912 breast ultrasound images found that Light Gradient Boosting Machine (LightGBM), the algorithm proposed in their work, which has an accuracy of 99.86%, outperformed other algorithms including the K-Nearest Neighbour (KNN), and RF in binary classification of cancerous cells as either malignant or benign. Similarly, Ara et al. in [8] used a ML techniques to develop model for classifying cancer cells into two main categories. Kumar et al. in [9] on the other hand focused on using ML ensemble techniques for breast cancer detection and classification. Their Optimized Stacking Ensemble Learning (OSEL) model showed a higher accuracy in performing the task than other ensemble ML techniques, such as Stochastic Gradient Boosting and XGBoost tested in their research. Ebrahim et al. in [10] tested eight predictive algorithms on National Cancer Institute dataset to identify which algorithm would predict cancer cell more accurately.

### B. *Selection of Algorithm*

LR, a linear model is a powerful predictive analysis tool that is especially useful for binary classification [11]. Rahman et al. [12] examined six ML algorithms for predicting Chronic Liver Disease (CLD) and LR algorithm was found to be the most effective in predicting CLD based on the selected features. Zhu et al. in [11] experimented with improved LR in the classification of binary variable and one or more independent variables to predict diabetes.

Likewise, Tree based algorithms including DT, RF and GB are widely researched with the intent of harnessing their strengths particularly in performing classification tasks. DT serve as foundational structures, offering transparency and interpretability by partitioning feature spaces into hierarchical branches thereby excelling in capturing non-linear relationships and feature interactions, enabling straightforward visualization of decision-making processes. Moving beyond individual trees, RF combines multiple DT through ensemble techniques, averting overfitting and increasing predictive accuracy [13]. By combining varied perspectives from individual trees, RF provides robust generalization and robustness to noisy data.

By extension, GB algorithm, a more advanced method, embraces an iterative refinement to enhance predictive performance and in particular, Gradient Boosting Trees, such as XGBoost. It employs sequential tree fitting to target the residuals of prior iterations, systematically improving model predictions. These algorithms perform better in modeling complex relationships, accommodating non-linearities, and excelling in predictive accuracy across domains [14][15]. These characteristics formed the basis on which we selected the algorithms in our study.

### C. *Train-Test Split*

For evaluation, datasets used in various studies are split into different proportions using the larger proportion to train algorithms while the smaller proportion is used to test at the inference stage of model development. In [10], the authors assessed the performance of some classical and deep learning algorithms used to predict breast cancer, including DT, LR, KNN, Support Vector Machine (SVM), Recurrent Neural Networks (RNN) and Ensemble Learning. They used Train/Test split of 70:30 and 90:10. DT and Ensemble methods showed higher accuracy both before and after feature selection. Whereas DT did not perform optimally in Kidney Cancer Lung Metastasis prediction as reported by [16] when trained with 52,222 data from Surveillance, Epidemiology, and End Results (SEER) database and 492 hospital patient data with Train/Test split of 70:30 returning accuracy of 82% which is significantly lower than in other studies reviewed.

### D. *Performance Metrics*

Efficient model development and deployment require rigorous assessment, evaluating the accuracy and other key metrics like precision, recall, and F1-score derived from the confusion matrix. Accuracy gauges correctly predicted instances against the total dataset, offering a general overview of predictive success. In imbalanced datasets, relying solely on accuracy can be deceptive. Therefore, other metrics such as precision, recall, and F1-score gain importance. Precision specifically gauges correctly predicted positive instances, which is crucial in scenarios like medical diagnoses where false positives can have adverse consequences. Recall assesses true positive predictions, essential for capturing all positive instances, especially critical in medical scenarios to avoid missing dangerous conditions. F1-score strikes a balance between precision and recall, offering a nuanced evaluation, particularly valuable when dealing with class imbalances in datasets.

These four metrics were assessed in our study (**TABLE 3**); they collectively provide a comprehensive assessment of a model's performance.

### E. *Datasets*

Data quality is fundamental in machine learning, shaping model development and real-world utility. The WDBC [6] has been pivotal in healthcare, especially for binary tumor classification, crucial in timely cancer detection and treatment planning. While studies like [17][18][19] employed smaller, open-source WDBC datasets (typically fewer than 600 records and 30 features), other studies in [10] and [15] diverged. For example, [10] used a substantial dataset from the National Cancer Institute (NIH) containing 1.7 million records and 210 features. Despite its size, dataset quality, marked by precision and representativeness, significantly influences outcomes. Smaller datasets with these qualities outperform larger, noisier ones. This distinction is evident in accuracy rates, with open-source datasets achieving 99.12%, 99.67%, and 100%, compared to the model in [10] with a lower accuracy of 98.7%.

### F. *Implementation Platform*

KNIME Analytics, a no-code tool recognized for its user-friendly interface and compatibility with various other tools, has been utilized for comprehensive ML research, as demonstrated in studies like [20] which looks at cancer incidence among individuals with HIV in Zimbabwe. Meanwhile, Python, with its extensive ecosystem and libraries like SciKit-Learn, has gained prominence in machine learning. Studies in [16][21][22] performed their cancer research work using Python. Both platforms have strong support from scientists, underlining the need for further research into their respective impacts on algorithm performance.

The findings of the literature are summarized in **TABLE 1**. The table highlights the latest studies that used ML techniques in cancer research, the data source used, train – test split ratio adopted in the study, the implementation platform used, the algorithm type and the model accuracy (a '–' has been used in the table in the case where the information was missing in the literature).

The recent surge in research on ML applications in healthcare, specifically in diverse cancer data sets, is evident. Nevertheless, a significant research gap persists concerning the impact of implementation platforms on algorithm performance in cancer classification.

While several studies have used different implementation platforms in developing ML models for predictive and classification tasks, none, to the best of our knowledge, have examined the impact of implementation platforms on ML algorithm performance. This gap forms the focal point of our research contribution, that will be explored in subsequent sections, highlighting the novelty and importance of our investigation.

TABLE 1. COMPARATIVE REVIEW OF SOME STUDIES THAT USED MACHINE LEARNING TECHNIQUES IN CANCER RESEARCH.

| Author, Year | Data Source | No of Records /Features | Train/Test Split | Implementation Platform | Algorithm Type | Model Accuracy |
|---|---|---|---|---|---|---|
| Ebrahim et al. [10],2023 | National Cancer Institute (NIH), USA | 70,079/107 | 70:30 &90:10 | Python | DT, LR, VM, LD, ET, KNN | 98.7% |
| Shafique et al.[18],2023 | Kaggle | 569/30 | 75:25 | - | RF, VM, GBM, LR, MLP, KNN | 100% |
| Uddin et al. [19], 2023 | UCI | 569/30 | 70:30 | Python | SVM, RF, KNN, NB, DT, LR, AB, GB, MLP, NCC, VC | 98.7% |
| Zhang et al [23]., 2022 | TCGA | 604/ - | - | R & Python | RF, SVM, libD3C | 99.67% |
| Aamir et.al.[24], 2022 | UCI | 569/26 | 80:20 &70:30 | Python & Tensor Flow | RF, GB, SVM, ANN, MLP | 99.12% |
| Yi et al., [16],2023 | SEER& Southwest Hospital, China. | 52,714 / - | 70:30 | Python | LR, XGB, RF, SVM, ANN, DT | - |
| Mahesh et al., [22],2022 | Kaggle | 143/10 | 70:30 | Python | NB, AltDT, RedEPT, RF | 98.20% |
| ATEŞ et al. [25] 2021 | Kaggle | 569/30 | 70:30 | Knime | NB, DT, MLP | 96.5% |
| Minnoor et al.[13] 2023 | UCI | 569/30 | 80:20 | - | RF, SVM, DT, MLP, KNN | 100% |
| Ara et al. [8] , 2021 | UCI | 569/30 | 75:25 | - | SVM, LR, KNN, DT, NB, RF | 96.5% |
| Liu, et al. [26]2018 | UCI | 569/30 | 75:25 | Python | LR | 96.5% |

Addressing the gap identified in the literature, the next section presents the methodology carried out.

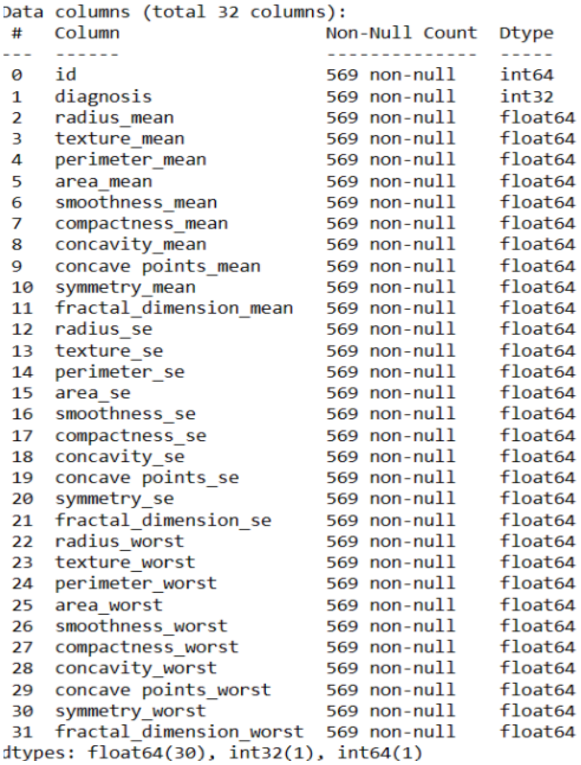
## III. METHODOLOGY

This study's methodology comprises systematic steps for a comparative analysis of ML algorithms using the WDBC dataset and two implementation platforms. The process as illustrated in  includes data collection, exploration, feature engineering, and selection using filtering and random forest techniques. The dataset was split into an 80% training set and a 20% test set before model development, ensuring a robust evaluation process.

### A. *Data Collection and Preprocessing*

We selected a publicly available dataset on UCI Machine Learning repository, the WDBC [6] because it was sourced from a medical research study and its extensive use in breast cancer machine learning research due to its real-world applicability, in addition to its popularity within the research community for binary classification task. With 569 occurrences and 30 attributes (benign tumours made up 62.7% of the total instances while the cancerous tumour, malignant class comprise 37.3%) was extracted from digitized Breast Mass Fine Needle Aspiration (FNA) specimens, including features like "Diagnosis" (categorized as Malignant (M) or Benign (B)) and various measurements from cell nuclei in biopsy images ("radius_mean," "texture_mean," "perimeter_mean," etc.) [6], providing a rich foundation for cancer predictive analysis.

TABLE 2. WDBC DATASET VARIABLES DATATYPE.



```
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   id                       569 non-null    int64  
 1   diagnosis                569 non-null    int32  
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int32(1), int64(1)
```

In the data preprocessing phase, the dataset was structured into a Python dataframe named "breast". The data was subsequently queried to ascertain the data types and to check for presence of any null values. According to TABLE 2, data consists of both integer and floating-point values, and no null values were found. Further analysis involved identifying outliers through box plots and the Capping method was applied to mitigate their impact. This technique, as presented by [27] involved setting values below the lower whisker to the lower whisker's value and values above the upper whisker to the upper whisker's value, ensuring an unbiased model.

Normalization was achieved through Z-Score Normalization (Standardization) which rescales each feature to normal distribution with a mean of 0 and a standard deviation of 1 [28][29]. Standardizing features to the same scale are essential to prevent algorithms from giving undue importance to larger-magnitude features, thus preserving fairness and accuracy across diverse ML algorithms. This process ensured that each feature contributed proportionally to the learning process, averting dominance by any single feature, and promoting balanced model decisions. Equation 1 below represents the computation formula for z-score standardization [29].

$$Z=(x-\mu)/\sigma. \qquad (1)$$

where z is the scaled value of the feature,
x is the original value of the feature,
μ is the mean value of the feature, and
σ is the standard deviation of the feature.

Correlation analysis was conducted to evaluate the relationship between each feature, a crucial step preceding feature selection, providing insights into features independently related to the target variable. This analysis was followed by a detailed examination of individual feature relationships, discerning the impact of changes in one feature on another and identifying strongly correlated independent features. High correlation between features suggests redundancy, potentially diminishing their value in the model, thus ensuring more effective predictions.

### *B. Feature Selection*

Selection of essential features is a crucial stage [30]. We employed both the Filter Method as in [4], and the Tree-Based Method as in [31]. Initially, the Filter Method was utilized to evaluate dataset features based on their correlation scores with the target variable. Features with coefficients $\leqslant 0.5$ were eliminated as they were considered to have low significance based on feature selection technique used in [32], while those above this threshold were retained, resulting in the identification of 15 out of the 30 features for further analysis. To confirm these selections, the Tree-Based Method was employed, utilizing the RF Classifier. This method, known for balancing interpretability and computational efficiency while capturing both linear and non-linear relationships between the features as shown in **Figure 2**, affirmed the chosen features, underscoring their significance in model development [30].

The synergy between the two methods ensured a comprehensive and accurate feature selection process, crucial for enhancing the model's predictive capabilities.

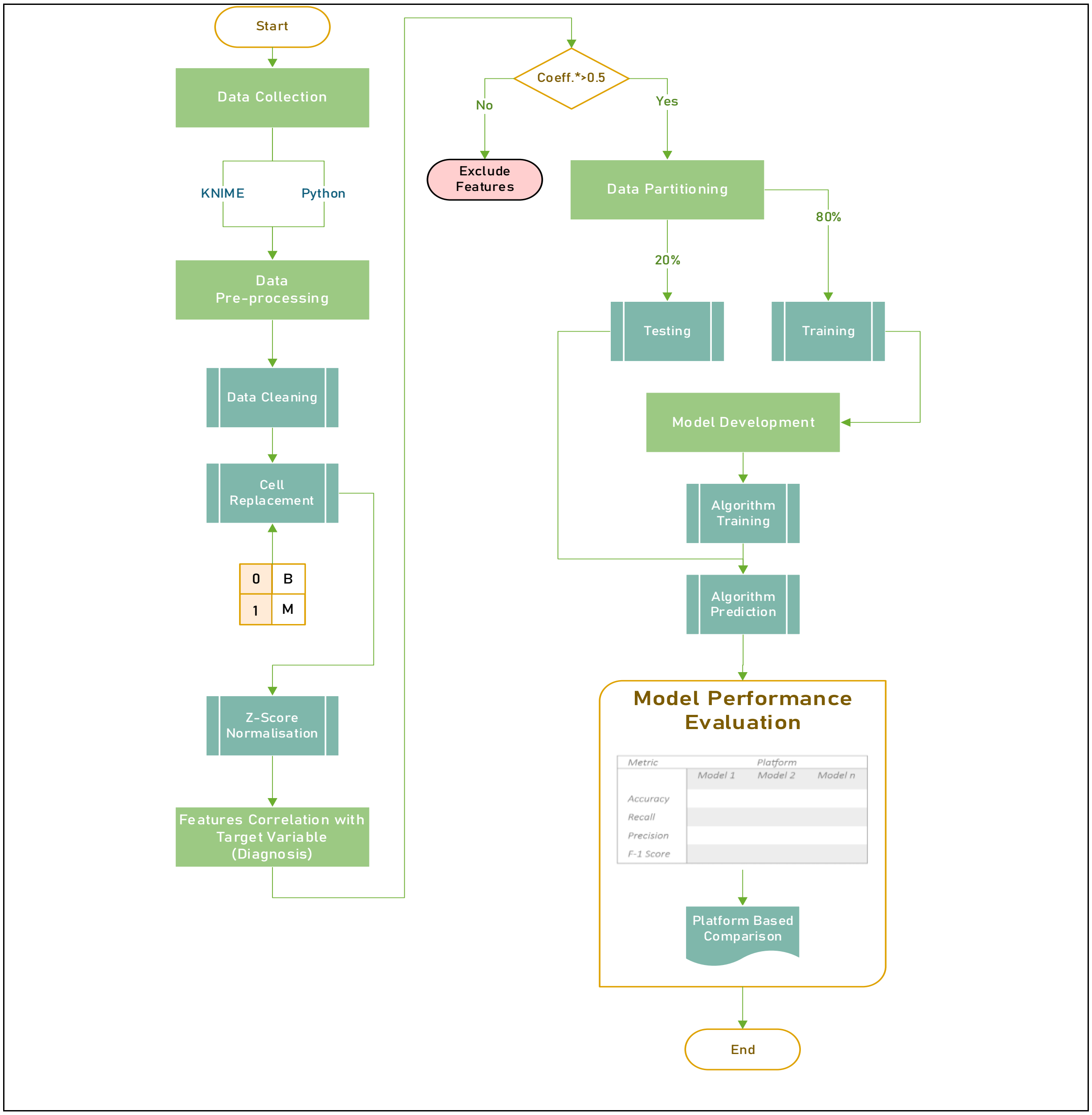


Figure 1. Flowchart illustrating the research methodology applied in this study.

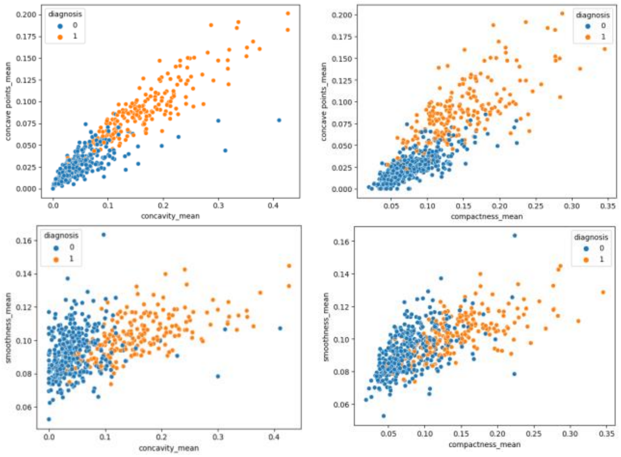


Figure 2. Scatter plot showing relationships between some of the features. (A view of relationships between some other features can be viewed on the GitHub [31]).

Understanding the relationship between the features helped to inform the class of ML algorithms that will be best suited for the classification task.

### *C Model Selection and Implementation*

Four Supervised ML classification algorithms were chosen, each based on their specific properties and extensive use in previous research. This study selected LR because of its ability to estimate outcome probabilities, along with its interpretability and computational efficiency. These attributes make LR a widely favoured option for binary classification tasks. DT, RF, and GB, all belonging to the Tree-Based algorithms category, were selected for their recursive partitioning approach, which efficiently identifies optimal features and split points, enhancing the models' accuracy.

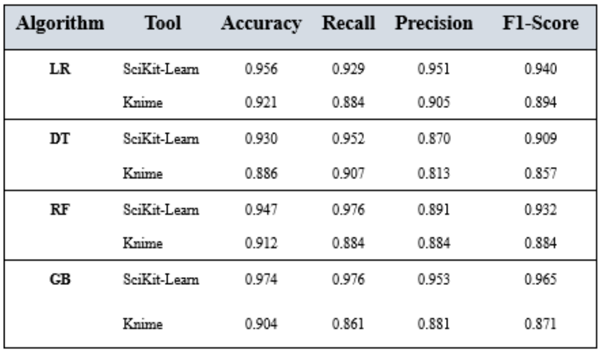
This study was carried out utilizing the Knime Analytics Platform Version 4.7.6 and Python version 3.11.4 (Jupyterlab) using the Scikit-Learn library. During this process, the algorithms underwent training and testing in their default configurations, with a maximum of 100 epochs, a learning rate of 0.1, and no parameter tuning—except in Knime, where the default split criterion for RF was adjusted from "Information Gain Ratio" to "Gini Index," aligning it with the default split criterion in Scikit Learn.

This adjustment was implemented to maintain fairness in the comparative evaluation. A train-test split ratio of 80:20 was applied, with 80% of the dataset allocated for training, enabling the algorithms to learn patterns, while the remaining 20% was reserved for testing, evaluating the models' ability to generalize to unseen data points. This methodology ensured a comprehensive evaluation of the algorithms' performance and their suitability for the classification task at hand. The source code (Python and Knime) and data for this study can be found in the public GitHub repository [31].

## IV. Results and Discussion

This section outlines the experimental results achieved following implementation of the four algorithms on both platforms comparatively in **TABLE 3** and visualized in **Figure 3** after assessing their Accuracy, Recall, Precision, and F1-Score.

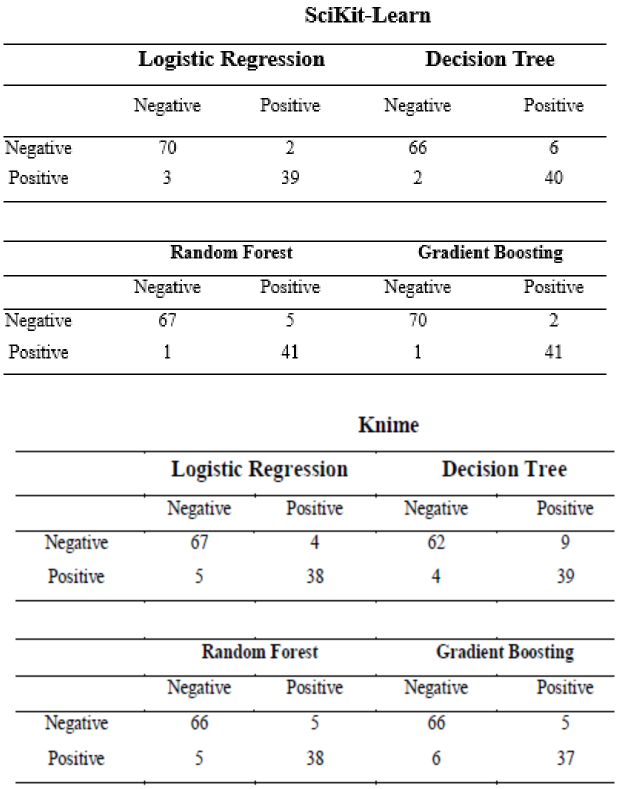
TABLE 3. COMPARATIVE ASSESSMENT OF MODEL PERFORMANCE ON THE TWO PLATFORMS.



| Algorithm | Tool | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|---|
| LR | SciKit-Learn | 0.956 | 0.929 | 0.951 | 0.940 |
| | Knime | 0.921 | 0.884 | 0.905 | 0.894 |
| DT | SciKit-Learn | 0.930 | 0.952 | 0.870 | 0.909 |
| | Knime | 0.886 | 0.907 | 0.813 | 0.857 |
| RF | SciKit-Learn | 0.947 | 0.976 | 0.891 | 0.932 |
| | Knime | 0.912 | 0.884 | 0.884 | 0.884 |
| GB | SciKit-Learn | 0.974 | 0.976 | 0.953 | 0.965 |
| | Knime | 0.904 | 0.861 | 0.881 | 0.871 |

Also, we reported the Confusion matrix, showing the True Positive, True Negative, False Positive, and False Negative values, providing a comprehensive evaluation of this study's outcomes in **TABLE 4**.

TABLE 4. PLATFORM BASED CONFUSION MATRIX OF THE ALGORITHMS.



**SciKit-Learn**

| | Logistic Regression | | Decision Tree | |
|---|---|---|---|---|
| | Negative | Positive | Negative | Positive |
| Negative | 70 | 2 | 66 | 6 |
| Positive | 3 | 39 | 2 | 40 |

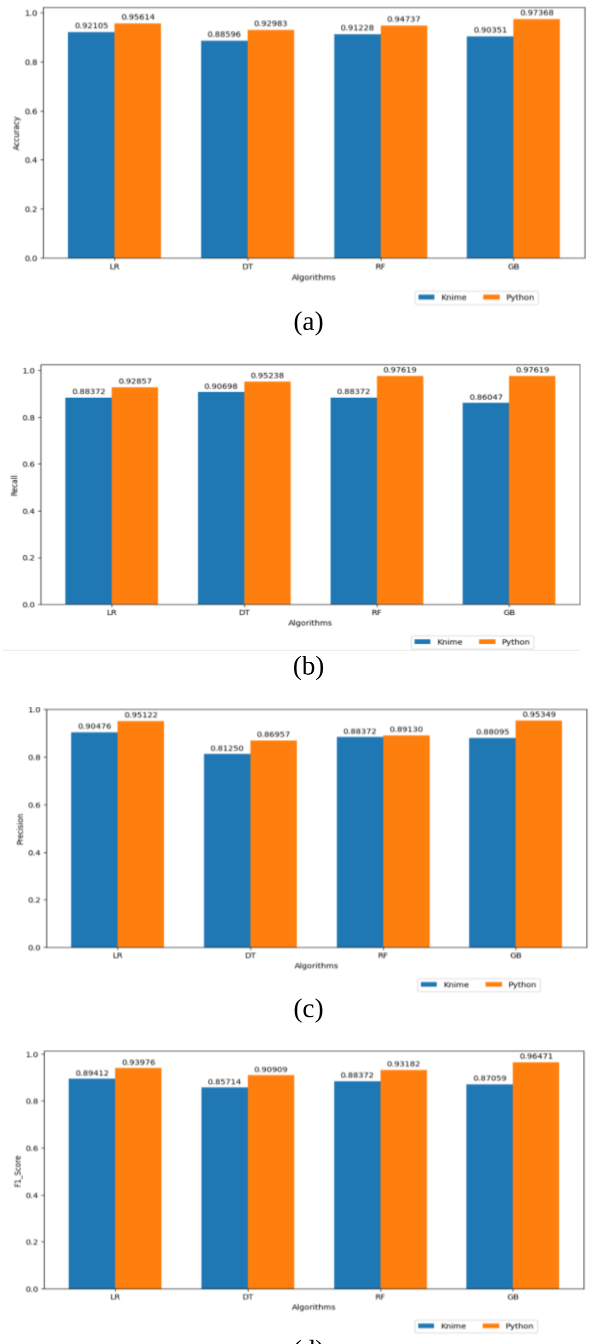
| | Random Forest | | Gradient Boosting | |
|---|---|---|---|---|
| | Negative | Positive | Negative | Positive |
| Negative | 67 | 5 | 70 | 2 |
| Positive | 1 | 41 | 1 | 41 |

.

**Knime**

| | Logistic Regression | | Decision Tree | |
|---|---|---|---|---|
| | Negative | Positive | Negative | Positive |
| Negative | 67 | 4 | 62 | 9 |
| Positive | 5 | 38 | 4 | 39 |

| | Random Forest | | Gradient Boosting | |
|---|---|---|---|---|
| | Negative | Positive | Negative | Positive |
| Negative | 66 | 5 | 66 | 5 |
| Positive | 5 | 38 | 6 | 37 |



(a)

(b)

(c)

(d)

Figure 3. Column Chart Visualization-Comparison of all the algorithms performance on both platforms for:
(a) Accuracy (b) Recall (c) Precision and (d) F1-Score.

In the KNIME Analytics platform, the LR algorithm achieved an Accuracy of 0.92105, with Recall, Precision, and F1 Score of 0.88372, 0.90476, and 0.89412, respectively signifying that the model correctly classified approximately 92.11% of the instances. In comparison, the DT algorithm demonstrated a slightly lower Accuracy of 0.88596, yet it exhibited higher Recall (0.90698) and F1 Score (0.85714), suggesting that it is proficient in capturing true positive instances while maintaining a balance between precision and recall, although its Precision score was 0.81250, indicating a relatively lower ability to avoid false positives.

The RF algorithm on the other hand achieved an Accuracy of 0.91228, almost on par with LR. It yielded Recall, Precision, and F1 Score of 0.88372, 0.88372, and 0.88372, respectively, presenting consistent performance across the metrics. The GB algorithm, like DT, secured an Accuracy of 0.90351, while it demonstrated a Recall of 0.86047, Precision of 0.88095, and F1 Score of 0.87059 reflecting a balanced trade-off between sensitivity and precision, critical in medical diagnosis scenarios.

However, on Python (Scikit-Learn) platform, the LR model exhibited superior performance, with an Accuracy of 0.95614. This shows an improvement in predictive accuracy when compared to its counterpart in KNIME Analytics. The Recall 0.92857, Precision 0.95122, and F1 Score 0.93976 further validate the model's proficiency in correctly classifying instances. DT and RF algorithms also displayed an improvement in their performance in the Python (Scikit-Learn) environment, with Accuracy values of 0.92981 and 0.94737, respectively.

Moreover, the Recall, Precision, and F1 Score values for these models witnessed an increase, thereby strengthening their overall predictive capabilities. The GB shows remarkable performance, attaining an Accuracy of 0.97368, a significant improvement compared to its counterpart in KNIME corroborating its impressive performance with Recall, Precision, and F1 Score values of 0.97619, 0.95349, and 0.96471, respectively, making it a standout in terms of all metrics.

The comparative analysis of these algorithms across the two platforms demonstrates the intricate relationship between algorithm choice, implementation environment, and resultant performance metrics. While KNIME Analytics rendered reliable results, Python (Scikit-Learn) emerged as the platform offering enhanced predictive accuracy across the board. Notably, the GB algorithm stood out in Python (Scikit-Learn), exhibiting remarkable performance, which is highly relevant in medical contexts where accurate classification holds paramount importance. These findings underscore the necessity of carefully considering both algorithm selection and platform for optimal performance in predictive modeling endeavors.

Additionally, the confusion matrix of the models was evaluated on their ability to predict both the 'Positive' and 'Negative' classes, and the calculated metrics offer valuable insights into their proficiency. The matrices revealed that while models generally perform well, some algorithms, such as DT and GB, consistently exhibit a higher number of True Positives emphasizing the accurate prediction of positive cases which is crucial in medical contexts to minimize the risk of false negatives.

Comparing the KNIME and SciKit-Learn platforms, a pattern emerges. Generally, the SciKit-Learn platform showcases slightly better performance metrics, particularly in terms of True Positives and True Negatives. This disparity suggests that the SciKit-Learn implementation may have certain advantages in terms of predictive accuracy and class separation.

Also in our analysis, we conducted a comparative assessment of the LR, GB, and RF models on scikit-learn against the Baseline Model Performance (BMP- available on UCI website) established using the same dataset from the UCI Machine Learning Repository. The LR and GB models demonstrated accuracy values of 95.6 and 97.4, respectively, falling within the BMP range [92.308-98.601]. Similarly, their precision scores (95.1 and 95.3) were consistent with the baseline range [91.555-98.576]. In contrast, the RF model reported accuracy and precision scores (94.7 and 89.1) below the lower limit of the baseline performance. On the other hand, for all metrics, the performances of the algorithms on Knime Analytics were lower than the lower limits of the BMP score.

## V. Conclusion and Future Work

This comparative experiment aimed to investigate the potential impact of machine learning implementation platform on the performance of machine learning models using the WDBC dataset and four classification algorithms during both training and inference phases in Python SciKit-Learn and Knime Analytics. The results demonstrated variation in the metrics for the algorithms in Python compared to Knime. While Knime showed its strength with the LR algorithm in terms of accuracy, Python presented different performance patterns, with DT excelling in recall and RF as well as GB providing high recall values, which are crucial in the context of cancer diagnosis as it suggests a reduced likelihood of false negatives.

These findings emphasize the significance of platform choice when considering the specific performance metrics required for a given application, shedding light on the intricate relationship between algorithm selection and the implementation environment. It is important to note that this study does not intend to render a verdict on the overall efficacy of either tool in ML model development but rather serves as an investigation into the potential disparities introduced by their respective architectures, providing insights for informed decision-making in predictive modeling endeavors.

Further research should explore a larger dataset as we hope this may contribute to the generalizability of the models and as a means of broadening the applicability of these findings. Future studies may also evaluate the performance of the algorithms on both platforms using other datasets. In addition, future work may:

(i) drill down to identify factors responsible for the observed differences by examining the platforms architecture,

(ii) extend the experiment by including some other classifiers algorithms, such as SVM and Multi-Layer Perceptron (MLP).

(iii) implement on different platforms including R and Weka or test multiple datasets.

## References

[1] B. S. Chhikara and K. Parang, “Global Cancer Statistics 2022: the trends projection analysis” *Chemical Biology Letters*, vol. 10, no. 1, Article no. 1, January. 2023, doi: https://scholar.google.com/scholar?q=urn:nbn:sciencein.cbl.2023.v10.451

[2] “*CANCER FACT SHEETS* - Global Cancer Observatory.” Accessed: Feb. 07, 2024. [Online]. Available: https://gco.iarc.fr/today/data/factsheets/cancers/20-Breast-fact-sheet.pdf

[3] V. D. P. Jasti et al., “Computational Technique Based on Machine Learning and Image Processing for Medical Image Analysis of Breast Cancer Diagnosis” *Security and Communication Networks*, vol. 2022, p.1-7, March. 2022, doi: 10.1155/2022/1918379.

[4] J. Kong et al., “Network-based machine learning approach to predict immunotherapy response in cancer patients” *Nature communications*, vol. 13, no. 1, Article no. 1, June 2022, doi: 10.1038/s41467-022-31535-6.

[5] W. Wolberg, O. Mangasarian, and W. Street, “Breast Cancer Wisconsin (Diagnostic).” UCI Machine Learning Repository, 1995. doi: 10.24432/C5DW2B.

[6] E. Michael, H. Ma, H. Li, and S. Qi, “An Optimized Framework for Breast Cancer Classification Using Machine Learning” BioMed Research International, vol. 2022, p. e8482022, Feb. 2022, doi: 10.1155/2022/8482022.

[7] S. Ara, A. Das, and A. Dey, “Malignant and Benign Breast Cancer Classification using Machine Learning Algorithms” in 2021 International Conference on Artificial Intelligence (ICAI), Apr. 2021, pp. 97–101. doi: 10.1109/ICAI52203.2021.9445249.

[8] M. Kumar, S. Singhal, S. Shekhar, B. Sharma, and G. Srivastava, “Optimized Stacking Ensemble Learning Model for Breast Cancer Detection and Classification Using Machine Learning” *Sustainability*, vol. 14, no. 21, Aricle. no. 21, January. 2022, doi: 10.3390/su142113998.

[9] M. Ebrahim, A. A. H. Sedky, and S. Mesbah, “Accuracy Assessment of Machine Learning Algorithms Used to Predict Breast Cancer” *Data*, vol. 8, no. 2, Article no. 2, Feb. 2023, doi: 10.3390/data8020035.

[10] C. Zhu, C. U. Idemudia, and W. Feng, “Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques” *Informatics in Medicine Unlocked*, vol. 17, p. 100179, January 2019, doi: 10.1016/j.imu.2019.100179.

[11] A. K. M. Rahman, F. M. Shamrat, Z. Tasnim, J. Roy, and S. Hossain, “A Comparative Study On Liver Disease Prediction Using Supervised Machine Learning Algorithms”. International Journal of Scientific & Technology Research. vol. 8, pp. 419–422, Nov. 2019.

[12] M. Minnoor and V. Baths, “Diagnosis of Breast Cancer Using Random Forests” *Procedia Computer Science*, vol. 218, pp. 429–437, Jan. 2023, doi: 10.1016/j.procs.2023.01.025.

**ISBN: 978-1-68558-136-7**

[13] W. Li, Y. Yin, X. Quan, and H. Zhang, “Gene Expression Value Prediction Based on XGBoost Algorithm” *Frontiers in Genetics*, vol. 10, 2019, Accessed: Feb. 07, 2024. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fgene.2019.01077

[14] X. Wan, “Influence of feature scaling on convergence of gradient iterative algorithm”. *Journal of physics: Conference series*., vol. 1213, no. 3, p. 032021, Jun. 2019, doi: 10.1088/1742-6596/1213/3/032021.

[15] X. Yi et al., “Development and External Validation of Machine Learning-Based Models for Predicting Lung Metastasis in Kidney Cancer: A Large Population-Based Study” *International Journal of Clinical Practice*, vol. 2023, p. e8001899, Jun. 2023, doi: 10.1155/2023/8001899.

[16] D. A. Omondiagbe, S. Veeramani, and A. S. Sidhu, “Machine Learning Classification Techniques for Breast Cancer Diagnosis” *IOP Conference Series: Materials Science and Engineering*, vol. 495, no. 1, p. 012033, Apr. 2019, doi: 10.1088/1757-899X/495/1/012033.

[17] R. Shafique et al., “Breast Cancer Prediction Using Fine Needle Aspiration Features and Upsampling with Supervised Machine Learning” *Cancers*, vol. 15, no. 3, Art. no. 3, Jan. 2023, doi: 10.3390/cancers15030681.

[18] K. M. M. Uddin, N. Biswas, S. T. Rikta, and S. K. Dey, “Machine learning-based diagnosis of breast cancer utilizing feature optimization technique” *Computer Methods and Programs in Biomedicine Update*, vol. 3, p. 100098, Jan. 2023, doi: 10.1016/j.cmpbup.2023.100098.

[19] T. Shamu et al., “Cancer incidence among people living with HIV in Zimbabwe: A record linkage study” *Cancer Reports*, vol. 5, no. 10, p. e1597, 2022, doi: 10.1002/cnr2.1597.

[20] Q. T. N. Nguyen et al., “Machine learning approaches for predicting 5-year breast cancer survival: A multicenter study” *Cancer Science*, vol. 114, no. 10, pp. 4063–4072, 2023, doi: 10.1111/cas.15917.

[21] T. R. Mahesh, V. Vinoth Kumar, V. Muthukumaran, H. K. Shashikala, B. Swapna, and S. Guluwadi, “Performance Analysis of XGBoost Ensemble Methods for Survivability with the Classification of Breast Cancer” *Journal of Sensors*, vol. 2022, p. e4649510, Sep. 2022, doi: 10.1155/2022/4649510.

[22] Y. Zhang et al., “Machine learning-based prognostic and metastasis models of kidney cancer” *Cancer Innovation*, vol. 1, no. 2, pp. 124–134, 2022, doi: 10.1002/cai2.22.

[23] S. Aamir et al., “Predicting Breast Cancer Leveraging Supervised Machine Learning Techniques” *Computational and Mathematical Methods in Medicine*, vol. 2022, p. e5869529, Aug. 2022, doi: 10.1155/2022/5869529.

[24] İ. Ateş and T. T. Bilgin, “The Investigation of the Success of Different Machine Learning Methods in Breast Cancer Diagnosis” *Konuralp Medical Journal*, vol. 13, no. 2, Article. no. 2, Jun. 2021, doi: 10.18521/ktd.912462.

[25] L. Liu, “Research on Logistic Regression Algorithm of Breast Cancer Diagnose Data by Machine Learning” in 2018 International Conference on Robots & Intelligent System (ICRIS), May 2018, pp. 157–160. doi: 10.1109/ICRIS.2018.00049

[26] X. Feng, Y. Cai, and R. Xin, “Optimizing diabetes classification with a machine learning-based framework” *BMC Bioinformatics*, vol. 24, no. 1, p. 428, Nov. 2023, doi: 10.1186/s12859-023-05467-x.

[27] S. Sumin, “The Impact of Z-Score Transformation Scaling on the Validity, Reliability, and Measurement Error of Instrument SATS-36,” JP3I (Jurnal Pengukuran Psikologi dan Pendidikan Indonesia), vol. 11, no. 2, Art. no. 2, Nov. 2022.

[28] M. Pagan, M. Zarlis, and A. Candra, “Investigating the impact of data scaling on the k-nearest neighbor algorithm” *Computer Science and Information Technologies*, vol. 4, no. 2, Art. no. 2, Jul. 2023, doi: 10.11591/csit.v4i2.p135-142.

[29] J. Cai, J. Luo, S. Wang, and S. Yang, “Feature selection in machine learning: A new perspective” *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018, doi: 10.1016/j.neucom.2017.11.077.

[30] G. Alfian et al., “Predicting Breast Cancer from Risk Factors Using SVM and Extra-Trees-Based Feature Selection Method” *Computers*, vol. 11, no. 9, Art. no. 9, Sep. 2022, doi: 10.3390/computers11090136.

[31] A.Olowolayemo (2023), Cancer3IPMLM, GitHub: https://github.com/ProfDee92/Cancer-3IPMLM/blob/main/README.md

[32] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O’Sullivan, “A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction” *Frontiers in Bioinformatics*, vol. 2, p. 927312, 2022.