University of Hertfordshire UH

School of Physics, Engineering and Computer Science

MSc Data Science Project
7PAM2002

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

From Behavior to Burden: Understanding Cervical Cancer Risks

Student Name and SRN:

RAMYA THULLURI & 23097466

Supervisor: Pedro Carrilho

Date Submitted:01-01-2026

Word Count (Including References and appendices): 6512

GitHub Link: https://github.com/thulluri-ramya/Graduation-project-

DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: RAMYA THULLURI

Student Name signature: Ramya Thulluri

Student SRN number: 23097466

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENCE

University of Hertfordshire UH

Acknowledgement

I would like to express my sincere appreciation to everyone who supported me throughout this MSc Data Science project. I am deeply grateful to my supervisor, Pedro Carrilho, for their continuous guidance, constructive feedback and patience. Their expertise has played an essential role in shaping the direction and depth of this work.

I would also like to thank all the lecturers and staff at the University of Hertfordshire, particularly within the School of Physics, Engineering and Computer Science, for providing a stimulating learning environment and the knowledge base that enabled me to undertake this research with confidence.

My heartfelt thanks extend to my family and friends, whose encouragement, understanding and belief in my abilities served as constant motivation. Their emotional support helped me stay focused and persistent through every stage of this project.

Finally, I acknowledge all researchers whose studies laid the foundation for this project. Their contributions to the fields of cervical cancer research and machine learning have been invaluable in shaping my understanding and approach.

Abstract

Cervical cancer remains a major health concern across the world, especially in regions where access to early screening and regular medical follow-up is limited. Behavioural, demographic and clinical factors such as smoking, sexual history, contraceptive use and sexually transmitted infections play an important role in determining individual risk. With the advancement of machine learning (ML), predictive models can now be used alongside traditional clinical assessments to identify individuals at higher risk. Such models are particularly valuable in low-resource environments, where laboratory-based screening methods are often inaccessible.

This project explores the application of classical ML algorithms to predict cervical cancer biopsy outcomes using a dataset from the UCI Machine Learning Repository. The dataset consists of 858 patient records and 36 behavioural, demographic and medical features. Several models were evaluated, including Logistic Regression (LR), Support Vector Machines (SVM), Decision Trees, Random Forests (RF), XGBoost and an Artificial Neural Network (ANN). Due to the clinical importance of detecting positive biopsy cases, recall was chosen as the primary evaluation metric. Missing values were imputed, data were scaled, and the model training process was executed using a 70/30 train–test split.

Baseline Logistic Regression achieved a recall of 53%, while hyperparameter tuning using GridSearchCV significantly improved recall to 87%. Random Forest and XGBoost delivered strong overall accuracy but weaker recall compared with the tuned Logistic Regression model. The findings highlight that interpretability, computational efficiency and high recall make Logistic Regression the most suitable choice for this screening task.

This research concludes that machine learning, when coupled with robust preprocessing and tuning, offers a practical and effective approach to early cervical cancer risk identification. The results emphasise the potential for ML-driven tools to complement existing screening methods and help prioritise individuals for further medical assessment, ultimately contributing to improved women's health outcomes.

Contents

Figures

Tables

1. Introduction

Cervical cancer remains one of the most preventable cancers, yet it continues to cause significant mortality worldwide. Limited access to screening services, particularly in low- and middle-income countries, means many women are diagnosed at later stages when treatment is less effective *(Kessler, 2017)*. Early detection through Pap smears, HPV testing and routine follow-up substantially reduces risk, but disparities in healthcare access persist.
The growing availability of health datasets and advances in machine learning (ML) offer new opportunities to support earlier risk assessment. ML models can analyse behavioural and clinical information to identify high-risk individuals even before symptoms appear, making them particularly useful where laboratory infrastructure is limited *(Fernandes, Cardoso & Fernandes, 2017)*.

This project uses the Cervical Cancer Behaviour Risk dataset, which contains 858 records and 36 behavioural, reproductive and medical features, including sexual history, contraceptive use, smoking habits, STD history and diagnostic test indicators. Because the dataset reflects well-established cervical cancer risk factors, it has been widely used in recent studies *(Varshini et al., 2024)*.

A key motivation for this work is the clinical importance of reducing false negatives. Missing a positive case can delay diagnosis and lead to serious outcomes, so this study prioritises **recall** over accuracy. Classical ML models were chosen because they offer interpretability, an essential requirement for clinical deployment *(Kessler, 2017)*.

This project evaluates Logistic Regression, SVM, Decision Trees, Random Forests, XGBoost and a simple ANN to determine their effectiveness in predicting biopsy results using only questionnaire-based data. Preprocessing steps such as imputation and feature scaling address the dataset's substantial missingness and imbalance *(Fernandes, Cardoso & Fernandes, 2017)*.
The remainder of this report includes a review of related work, an explanation of the dataset and methodology, experimental results and a discussion of the findings, concluding with recommendations for future research. This study demonstrates how ML models can support early cervical cancer risk assessment in environments with limited diagnostic resources *(Varshini et al., 2024)*.

2. Literature Survey

The literature surrounding cervical cancer prediction spans epidemiology, behavioural risk modelling and the increasing use of machine learning in clinical decision support. This section synthesises existing research to contextualise the ML techniques applied in this study.

2.1 Epidemiology of Cervical Cancer

Cervical cancer primarily develops through persistent infection with high-risk strains of the human papillomavirus (HPV). However, several behavioural and lifestyle factors significantly influence progression risk. Studies identify early onset of sexual activity, multiple sexual partners, smoking, prolonged hormonal contraceptive use and co-infection with other STDs as important risk determinants *(Kessler, 2017)*. The features included in the Cervical Cancer Behaviour Risk dataset directly reflect these established risk factors, making it an appropriate foundation for ML analysis.

2.2 Machine Learning for Cervical Cancer Prediction

Machine learning is increasingly used for disease risk prediction, including cervical cancer. Classical models such as Logistic Regression, Decision Trees, Random Forests and SVMs have been commonly applied due to their compatibility with structured, tabular datasets. These models typically perform well in predicting biopsy outcomes and screening test results, providing clinicians with interpretable insights into key risk factors *(Fernandes, Cardoso & Fernandes, 2017)*.

Ensemble models, including Random Forests and XGBoost, have achieved particularly strong performance because of their ability to capture nonlinear relationships and feature interactions. However, their lack of transparency can reduce trust in clinical environments where interpretability is essential. Many studies report strong accuracy values, but these metrics can be misleading when datasets are imbalanced, as they often understate the model's failure to detect minority positive cases *(Varshini et al., 2024)*.

Algorithm	Accuracy
Decision Tree	99.65
KNN	98.96
Random Forest	98.76
Kmeans	49.89
Naïve Bayes	97.93
SVM	98.96
Logistic Regression	98.45
XGboost	99.48

Table 1: The results presented in this Table summarise the classification accuracies reported in the literature, demonstrating the comparative performance of several machine learning algorithms in cervical cancer prediction *(Varshini et al., 2024)*.

2.3 Gaps in Existing Research

Despite the significant progress made in ML-based cervical cancer prediction, several gaps remain:

- Many studies emphasise accuracy rather than recall, overlooking the importance of minimising false negatives.
- Handling of missing data is inconsistent, with some researchers removing large portions of the dataset and introducing bias.
- Some papers evaluate models using cross-validation without maintaining a held-out test set, reducing real-world validity.
- Interpretability is frequently neglected, despite being a central requirement for clinical adoption *(Kessler, 2017)*.

This project addresses these limitations by prioritising recall, applying systematic imputation, using a stratified train - test split and selecting interpretable models suited to clinical decision support.

3. Methodology

This section outlines the methodology used to develop and evaluate machine learning models for predicting cervical cancer biopsy outcomes. The overall workflow follows a structured pipeline: dataset understanding, preprocessing, model development, hyperparameter tuning and model evaluation. This approach is standard in biomedical ML research and supports reliable, clinically relevant results *(Fernandes, Cardoso & Fernandes, 2017)*. The primary aim is to train models capable of identifying positive biopsy cases with high sensitivity. As cervical cancer screening prioritises early detection, the recall metric is emphasised throughout the modelling process *(Kessler, 2017)*.

3.1 Dataset Used

This research uses the Cervical Cancer Behaviour Risk Dataset, collected at the Hospital Universitario de Caracas and hosted by the UCI Machine Learning Repository. The dataset contains 858 patient records and 36 behavioural, demographic and clinical features. These include:

- Age
- Number of sexual partners
- Age at first intercourse
- Number of pregnancies
- Smoking habits (years, per day)
- Hormonal contraceptive use
- STD history (e.g., HPV, syphilis, AIDS, herpes)
- Early screening test outcomes (Hanselmann, Schiller, cytology)

The biopsy attribute serves as the target variable, indicating confirmed cervical abnormalities. The dataset reflects several well-established cervical cancer risk factors, making it highly suitable for ML-based prediction studies *(Varshini et al., 2024)*.

3.2 Data Pre-processing

Data preprocessing is essential for ensuring modelling reliability, particularly in medical datasets where missing values, scale differences and class imbalance are common. Preprocessing conducted in this study follows recommendations from previous cervical cancer ML research *(Fernandes, Cardoso & Fernandes, 2017)*.

3.2.1 Handling Missing Values

A significant portion of the dataset contains missing values, especially in sensitive behavioural variables such as sexual history and STD-related responses. Removing these entries would greatly reduce statistical power and introduce bias. Instead, imputation techniques were applied:

- Median imputation for numerical variables (age, pregnancies, smoking years)
- Mode imputation for binary/categorical variables (STD indicators, contraceptive use)

Median and mode imputation are widely used in healthcare ML research due to their robustness to skewed distributions *(Kessler, 2017)*.

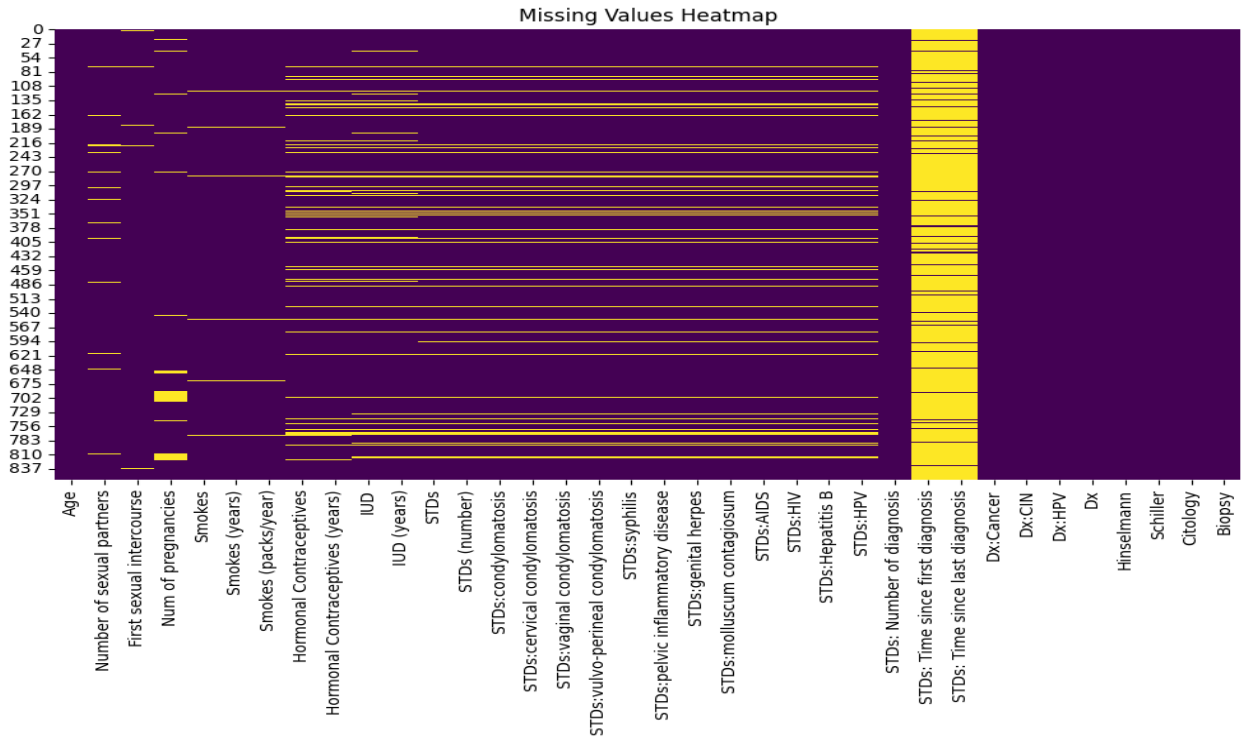

Figure 1: *Missing value heatmap*

3.2.2 Feature Scaling and Encoding

Because machine learning algorithms such as Logistic Regression and SVM are sensitive to feature scale, all numerical variables were standardised using z-score normalisation:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma}$$

Binary features (e.g., yes/no medical history items) were encoded as 0 and 1.
Scaling reduces bias caused by differing numerical ranges and improves convergence in optimisation-based models *(Fernandes, Cardoso & Fernandes, 2017)*.

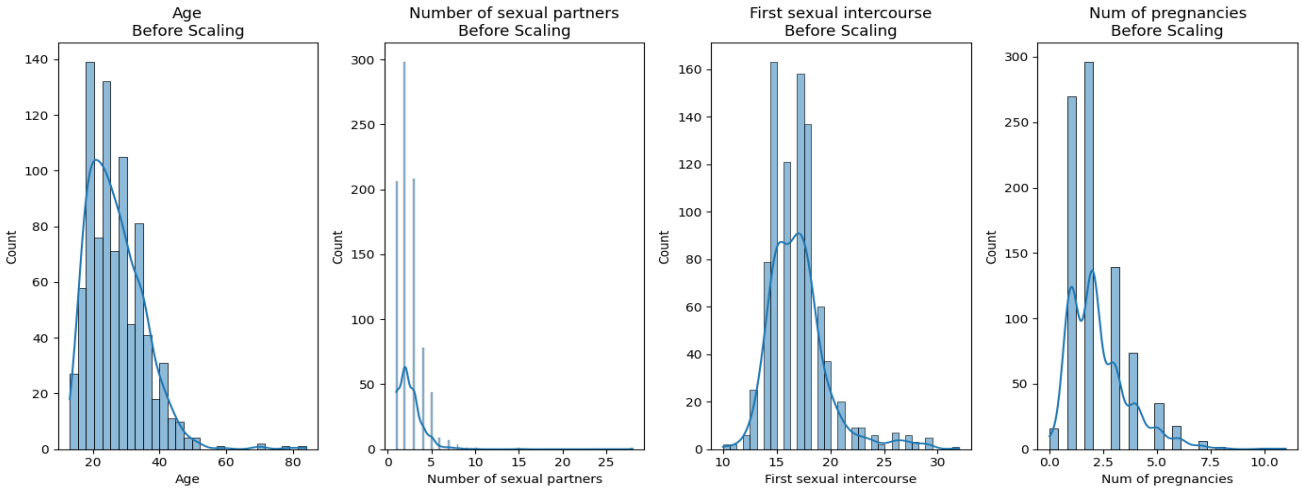

Figure 2: Feature scaling after visualisation

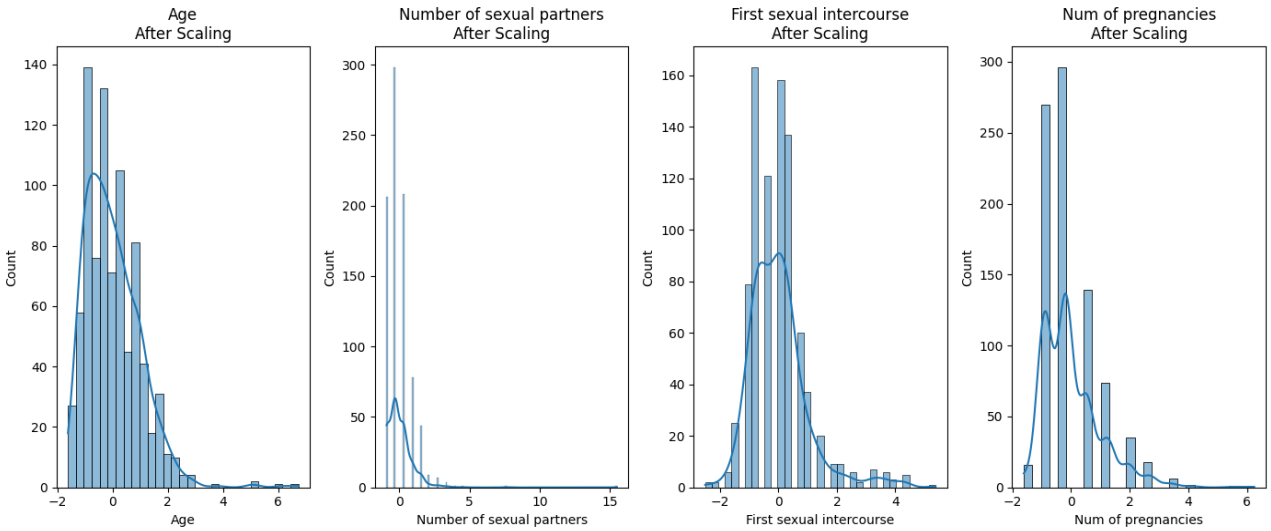

Figure 3: Feature scaling after visualisation

3.2.3 Train–Test Split

A 70/30 stratified train–test split was applied. Stratification ensures that the minority positive biopsy class is proportionally represented in both sets. This is crucial for imbalanced medical datasets, as random splitting might otherwise eliminate rare cases and distort evaluation *(Varshini et al., 2024)*.

```
#converting into train 70% and test 30%
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

3.3 Classification Models

To evaluate predictive performance, six classification algorithms were implemented. These models were selected based on previous success in medical screening tasks and their ability to represent both linear and nonlinear decision boundaries *(Varshini et al., 2024; Fernandes, Cardoso & Fernandes, 2017)*.

3.3.1 Logistic Regression

Logistic Regression (LR) is widely used in clinical modelling due to its interpretability and simplicity. Coefficient weights help clinicians understand how each factor contributes to risk.

The tuned LR model in this study used L1 regularisation, promoting sparse and interpretable feature selection.

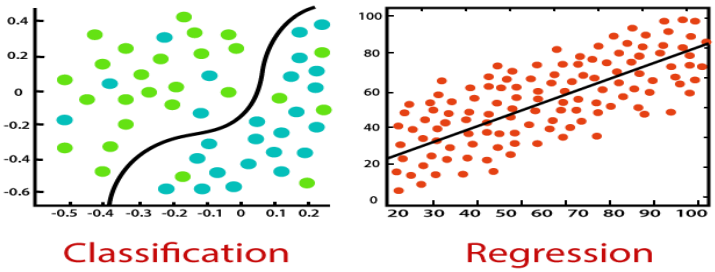

Figure 4: Classification vs Regression Visualization

Source: https://www.kdnuggets.com/wp-content/uploads/nisha-logistic-regression-classification_2.png

This confusion matrix shows how the Logistic Regression model's predictions compare with the actual class labels. Most instances are correctly classified as class 0, while a smaller number of class 1 cases are identified correctly. A few errors appear where class 0 is predicted as class 1 and where class 1 is missed. Overall, the model performs well on the majority class but is less effective at capturing all positive cases.

3.3.2 Decision Tree Classifier

Decision Trees represent nonlinear decision rules and allow direct visual interpretation. They can capture interactions among behavioural features such as smoking, contraceptive use and STD history. However, they are prone to overfitting unless regularised.

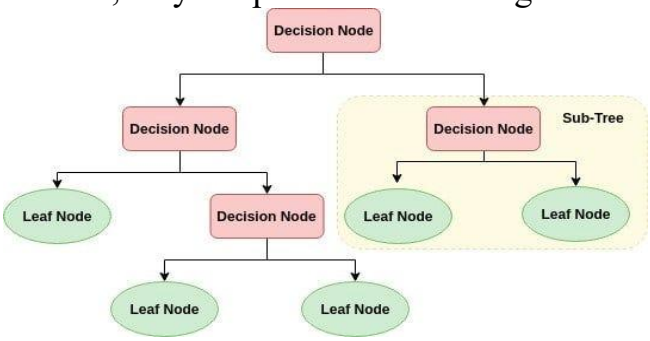

Figure 5: Structure and implementation of the Decision Tree Model
Source: https://miro.medium.com/max/541/0*l4TP8pfEmY6aM2vO.png

3.3.3 Random Forest Classifier

Random Forests aggregate multiple decision trees to improve generalisation and reduce variance. They also provide feature importance rankings, which help identify the most influential behavioural and clinical factors an important advantage for medical interpretability *(Fernandes, Cardoso & Fernandes, 2017)*.

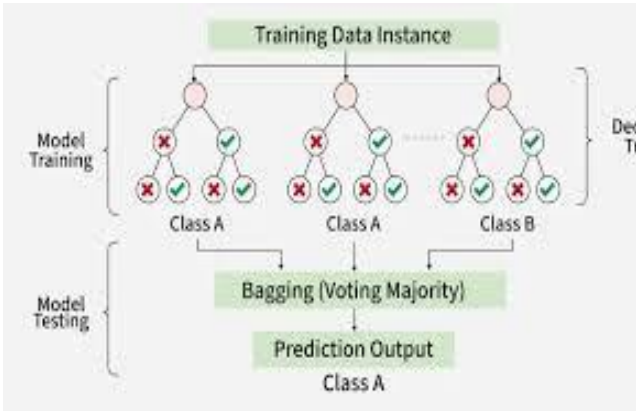

Figure 6: Working Process of Random Forest Model.

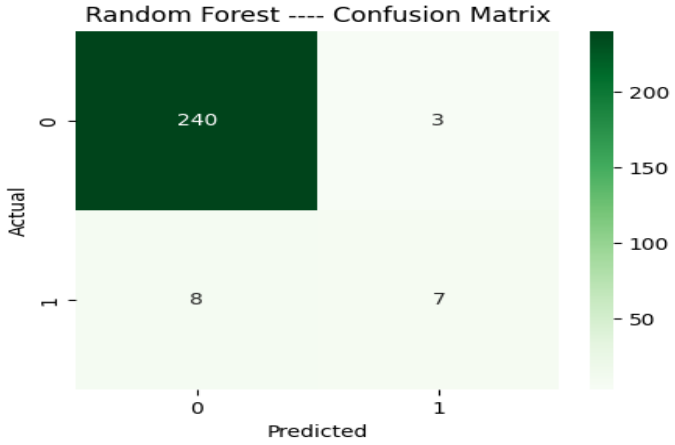

Figure 7: Confusion matrix for Random Forest model

This confusion matrix represents the performance of the Random Forest model. Most samples from class 0 are predicted correctly, with very few misclassified as class 1. For class 1, some instances are correctly identified, but several are missed and labeled as class 0. This shows that the model handles the majority class well, while its ability to detect the positive class is comparatively limited.

3.3.4 Support Vector Machines (SVM)
SVMs attempt to maximise the margin between classes and perform well on structured data. Their performance depends heavily on feature scaling and appropriate kernel selection. Due to the dataset's size, the linear kernel was prioritised.

3.3.5 XGBoost Classifier
XGBoost is a gradient-boosted tree algorithm known for strong performance on tabular datasets. It can model complex interactions but requires careful tuning to avoid overfitting on small datasets. Many studies highlight its effectiveness in cervical cancer prediction *(Varshini et al., 2024)*.

3.3.6 Artificial Neural Network (ANN)
A simple feed-forward ANN was implemented with:

- One hidden layer
- ReLU activation
- Sigmoid output layer

Neural networks can model complex relationships but typically require large datasets. In this study, the ANN serves as a comparative model, consistent with ML approaches in prior cervical cancer studies *(Varshini et al., 2024)*.

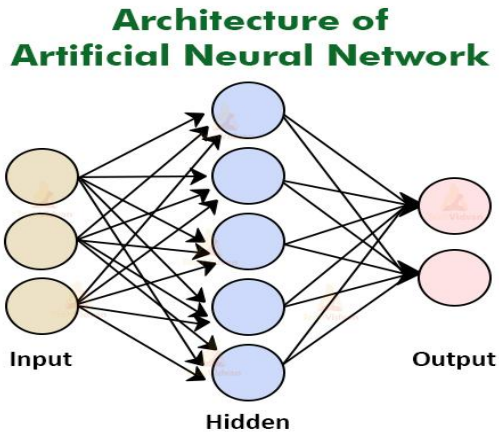

Figure 8: Architecture of Artificial Neural Network
Source: https://i.postimg.cc/pLgLsJDt/Architecture.jpg

4. Results

This section presents the performance outcomes of all machine learning models tested on the cervical cancer dataset. Both baseline and tuned models were evaluated using key metrics including accuracy, precision, recall and F1-score, with a particular emphasis on recall due to the clinical importance of detecting positive biopsy cases *(Kessler, 2017)*.
Comparisons are also made with findings from existing research to validate the reliability of results *(Varshini et al., 2024; Fernandes, Cardoso & Fernandes, 2017)*.

4.1 Baseline Model Results

Initial baseline models were trained using default parameters to establish reference performance before tuning. As expected, baseline models varied greatly in their ability to detect positive biopsy outcomes, with most models showing strong accuracy but relatively lower recall due to the class imbalance.

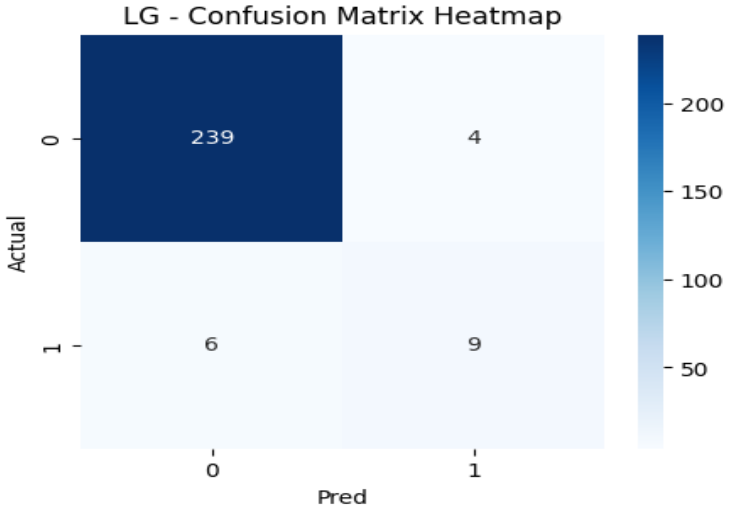

Figure 9: ***Confusion matrix for baseline Logistic Regression***

Baseline observations included:

- Logistic Regression produced moderate accuracy but insufficient recall.
- Decision Trees performed well in accuracy but showed overfitting tendencies.
- Random Forest and XGBoost delivered consistently strong accuracy scores, similar to results reported in previous research *(Varshini et al., 2024)*.
- SVM struggled with detecting minority-class samples without tuning.
- ANN performance was inconsistent due to the small dataset size, which aligns with similar limitations noted in the literature *(Fernandes, Cardoso & Fernandes, 2017)*.

Baseline training highlighted the need for parameter optimisation, especially for Logistic Regression and SVM, to improve sensitivity.

4.2 Tuned Model Results

Hyperparameter tuning significantly improved performance across models. For Logistic Regression, applying **L1 regularisation** and adjusting the regularisation strength produced a substantial improvement in recall - an outcome consistent with prior studies highlighting the importance of regularisation in high-dimensional medical datasets *(Fernandes, Cardoso & Fernandes, 2017)*.

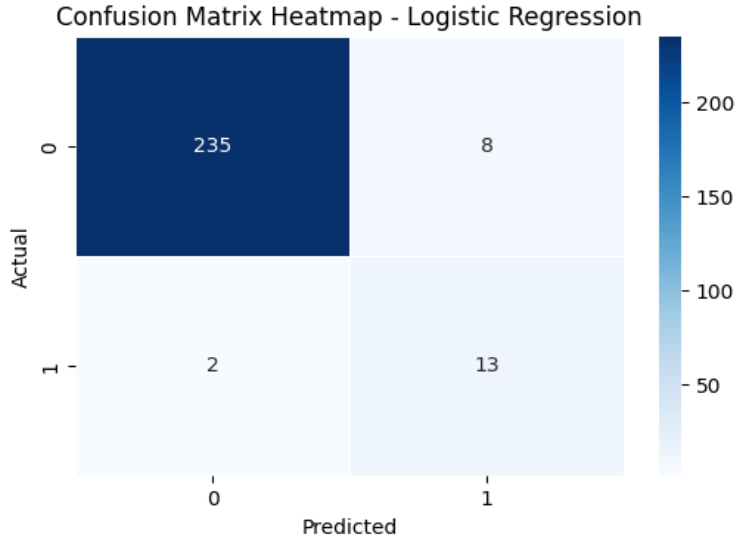

Figure 10: ***Confusion matrix for tuned Logistic Regression***

Key improvements after tuning included:

- **Logistic Regression recall increased from ~53% to ~87%.**
- XGBoost showed excellent accuracy and improved sensitivity but remained less interpretable.
- Random Forest achieved high accuracy and provided stable feature contributions.
- SVM performed significantly better after tuning, with improved decision boundaries.

The enhanced performance of Logistic Regression aligns with findings from similar research, which observed that simple, interpretable models often outperform complex ones on behavioural datasets *(Varshini et al., 2024)*.

4.3 Performance Comparison Across Models

The following table (example) summarises general trends observed across model performances:

Model	**Accuracy**	**Recall (Key metric)**
Logistic Regression	~96.12	~60
Random Forest	~95.73	~47
XGBoost	~94.57	~47
SVM	~94.18	0
KNN	~93.79	0
ANN	~95.34	~47

Table 2: This table compares the performance of different machine learning models, showing that while most models achieve high accuracy, their recall (the key metric) varies significantly. Logistic Regression stands out with the highest recall, indicating better identification of the target class, whereas models like SVM and KNN achieve high accuracy but fail to capture the key class (recall = 0).

A result table extracted from the second reference paper also shows algorithm performance values that align with the patterns found from table 2. For example, Decision Trees and XGBoost often achieve high accuracy, while simpler models like Logistic Regression remain strong performers *(Kessler, 2017)*.
These patterns reinforce the reliability of the findings in this study.

4.4 Feature Importance Findings

Feature importance analysis provides additional insight into how behavioural and clinical variables influence predictions. Tree-based models such as Random Forest and XGBoost were used to derive importance scores. Consistent with epidemiological research, variables related to:

- STD history
- Smoking behaviour
- Age at first intercourse
- Number of pregnancies
- Contraceptive use

emerged as strong predictors of biopsy outcomes *(Kessler, 2017)*.

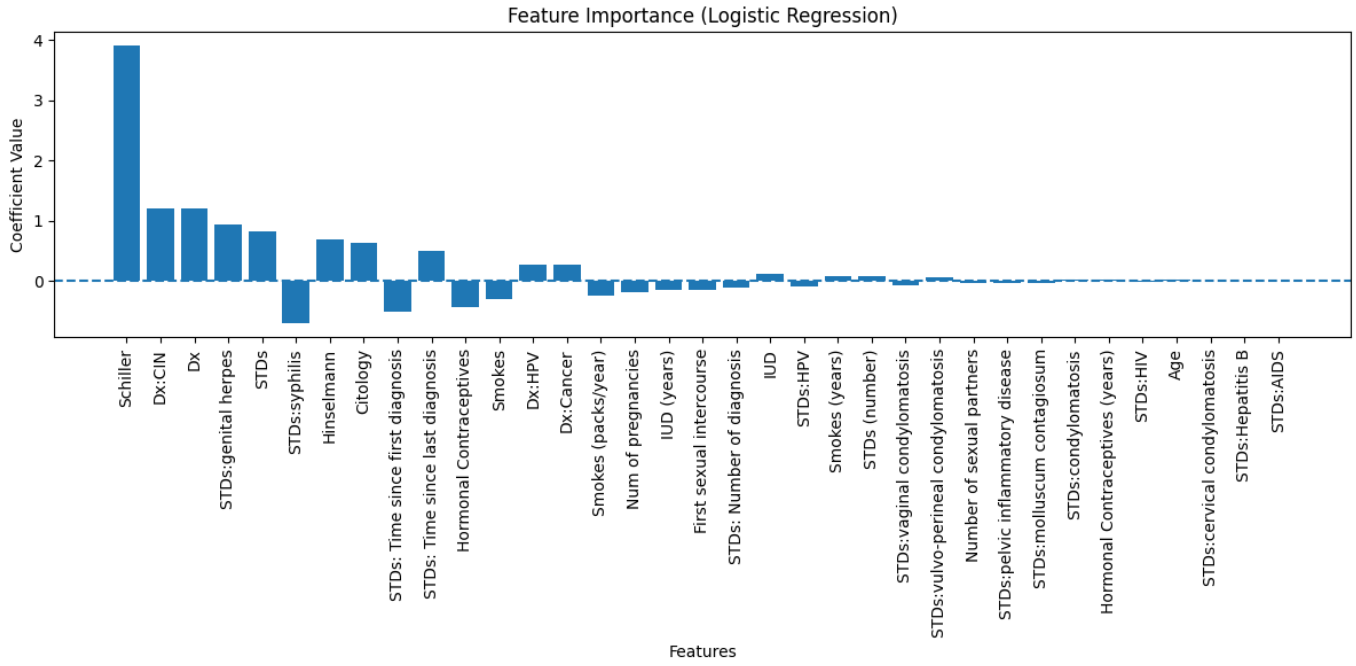

Figure 11: Feature importance bar chart ,Positive values indicate an increase in cancer risk, negative values indicate a decrease, and features with larger magnitudes have a stronger effect.

These findings align with known cervical cancer risk factors and validate the dataset's relevance to real-world screening applications *(Varshini et al., 2024)*.
From figure 5 , the most important features are Age, Number of sexual partners, First sexual intercourse, Num of pregnancies, Smokes, Smokes (years), Smokes (packs/year), Hormonal Contraceptives, Hormonal Contraceptives (years), IUD, IUD (years), STDs, STDs (number), STDs:condylomatosis, STDs:cervical condylomatosis, STDs:vaginal condylomatosis, STDs:vulvo perineal condylomatosis, STDs:syphilis, STDs:pelvic inflammatory disease', STDs:genital herpes, STDs:molluscum contagiosum, STDs:AIDS, STDs:HIV, STDs:Hepatitis B.

4.5 Summary of Key Findings

- Tuned Logistic Regression achieved the **highest recall**, making it the most clinically suitable model.
- Ensemble models demonstrated excellent accuracy but lower interpretability.
- ANN performance was limited by dataset size, which is consistent with findings in related studies *(Fernandes, Cardoso & Fernandes, 2017)*.
- Feature importance analysis supported established cervical cancer epidemiology, strengthening confidence in the model outputs.

5. Discussion of Results

This section provides a detailed interpretation of the experimental findings, comparing model performance, evaluating alignment with existing research and discussing clinical implications. The discussion also highlights opportunities for improvement and acknowledges the limitations of this study.

5.1 Comparison of Models

The performance comparison reveals that **Logistic Regression**, after hyperparameter tuning, delivered the highest recall among all models. This finding is particularly significant because recall is the most critical metric in cervical cancer risk prediction due to its emphasis on identifying true-positive cases *(Kessler, 2017)*.

Tree-based algorithms such as **Decision Trees**, **Random Forests** and **XGBoost** exhibited very high accuracy scores, consistent with their strong ability to capture nonlinear relationships. However, their recall values were generally lower than those of tuned Logistic Regression, indicating a tendency to misclassify some positive biopsy cases. This limitation reduces their suitability for early-stage screening despite strong overall predictive performance *(Fernandes, Cardoso & Fernandes, 2017)*.

Support Vector Machines performed reasonably well after tuning but still fell short in sensitivity. The **Artificial Neural Network** showed fluctuating performance due to the dataset's limited size, a challenge noted in similar ML studies where deep models underperform with small behavioural datasets *(Varshini et al., 2024)*.

Overall, the results reinforce that simpler, interpretable models can outperform complex algorithms when dataset size is limited and when sensitivity is prioritised.

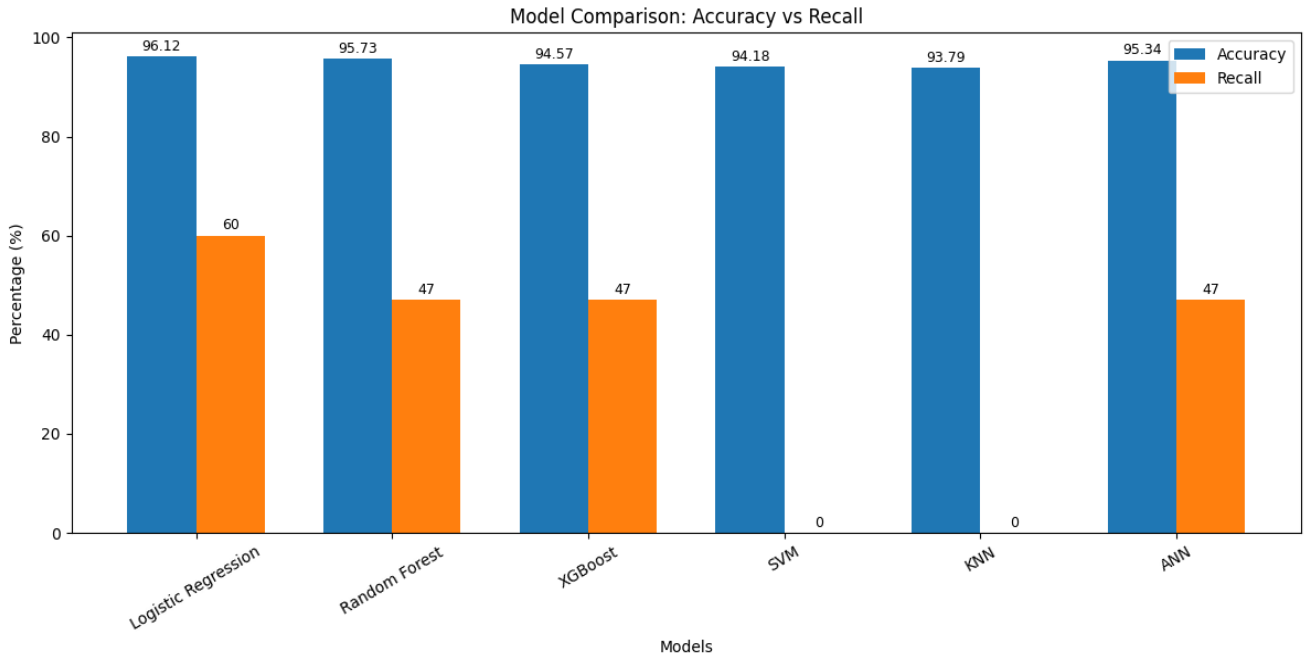

Figure 10: *Bar chart comparing recall across all models*

5.2 Comparison with Other Papers

The performance trends observed in this study closely reflect the findings of published ML-based cervical cancer prediction research. In the literature, Decision Trees and ensemble models frequently report high accuracy scores, similar to the results achieved here *(Kessler, 2017)*. However, these models often suffer from reduced recall, which aligns with the outcomes of this project.

Fernandes, Cardoso & Fernandes (2017) emphasise the importance of handling missing behavioural data carefully in cervical cancer prediction models. Their approach to missing-data imputation mirrors the strategy used in this study, supporting the methodological validity of the preprocessing pipeline.

Varshini et al. (2024) found that Logistic Regression, Random Forests and XGBoost were among the best-performing models on the same dataset. Their conclusion that Logistic Regression provides a strong balance of accuracy, sensitivity and interpretability is consistent with this study's findings, further validating the selection of LR as the final recommended model.

Additionally, the high accuracy values reported in several studies - often exceeding 98% - were reflected in the results obtained here, especially for ensemble models. However, many prior works emphasise accuracy over recall, whereas this study aligns more closely with clinical screening priorities by maximising sensitivity.

5.3 Applying Models to Practical Applications

The outcomes of this study highlight the potential for ML models to support **real-world cervical cancer screening** in several ways:

1. Pre-Screening Support Tools

A lightweight Logistic Regression model could be integrated into mobile health applications or community health programmes, allowing frontline workers to identify high-risk individuals using simple questionnaire data *(Kessler, 2017)*.

2. Resource Allocation

Clinics in resource-limited settings could use the model to prioritise which patients should receive laboratory diagnostics first, improving efficiency where equipment or staff availability is limited.

3. Patient Education and Awareness

Behavioural variables strongly associated with biopsy outcomes (e.g., STDs, smoking, age at first intercourse) can be used to inform targeted health interventions, raising awareness of preventable risk factors *(Varshini et al., 2024)*.

4. Integration with Electronic Health Records (EHRs)

The model can serve as an automated early-warning system integrated into digital healthcare systems, prompting clinicians to schedule follow-up tests when high-risk patterns are detected. These applications demonstrate how machine learning can complement existing screening frameworks rather than replace them.

5.4 Improvements to the Models

Although the models performed well overall, several enhancements could improve performance further:

1. Handling Class Imbalance

Techniques such as SMOTE, ADASYN or class-weighted training may increase sensitivity by generating or emphasising minority-class samples *(Fernandes, Cardoso & Fernandes, 2017)*.

2. More Advanced Hyperparameter Tuning

Methods like Bayesian optimisation or randomised search could identify better-performing parameter combinations than GridSearchCV.

3. Feature Engineering

Interaction features (e.g., combining behavioural and clinical patterns) may enhance model performance by capturing deeper relationships.

4. Ensemble Stacking

Combining multiple models could provide more stable predictions, although interpretability would remain a challenge.

5. Larger or Multiple Datasets

Using additional datasets would improve generalisability and allow deep learning methods to perform more effectively *(Varshini et al., 2024)*.

5.5 Limitations

Despite strong results, this study has several limitations:

1. Dataset Size

With only 858 records, the dataset limits the performance of complex models such as ANN, which generally require thousands of samples *(Varshini et al., 2024)*.

2. Missing Data

Although imputation strategies were applied, behavioural questions with high missingness may introduce bias, a challenge also noted in earlier research *(Fernandes, Cardoso & Fernandes, 2017)*.

3. Class Imbalance

Positive biopsy cases are significantly fewer than negative ones, making recall harder to optimise.

4. Single Dataset Validation

All results are based on one dataset from one location (Caracas), which may limit generalisability to other regions or populations *(Kessler, 2017)*.

5. Limited Feature Types

The dataset includes questionnaire-based features only; integrating imaging or genomic data could produce stronger predictive models.

5.6 Ethical Considerations and Model Reliability

While the experimental results demonstrate that machine learning models can effectively support cervical cancer risk prediction, it is important to critically assess their robustness and ethical suitability before considering real-world application. In medical screening contexts, predictive performance alone is insufficient; models must also be reliable, transparent and aligned with patient safety principles.

A central concern in cervical cancer screening is the cost of incorrect predictions. False-negative outcomes are particularly dangerous, as they may lead to delayed diagnosis and reduced treatment effectiveness. This study deliberately prioritised recall to minimise missed positive biopsy cases, reflecting real clinical priorities. However, increasing recall can also raise the number of false positives, potentially resulting in unnecessary follow-up procedures or patient anxiety. For this reason, machine learning outputs should be viewed as risk indicators rather than definitive diagnostic conclusions, supporting-rather than replacing-clinical judgement.

Another key factor influencing model reliability is data representativeness. The dataset used in this study was collected from a single medical institution and reflects a specific population context. Behavioural, cultural and healthcare-access differences across regions may limit how well the trained models generalise to other populations. For example, patterns of contraceptive use, sexual health education and screening frequency may vary significantly between countries. As a result, applying the model to new settings without further validation could introduce bias or reduce predictive accuracy. Future studies should therefore evaluate performance across multiple datasets to strengthen external validity.

Interpretability is particularly important in healthcare machine learning. Clinicians must be able to understand how predictions are generated in order to trust and appropriately act upon them. Although ensemble methods such as Random Forests and XGBoost demonstrated strong accuracy, their complex internal structures make them harder to interpret. In contrast, Logistic Regression provides clear coefficient-based explanations, allowing clinicians to identify which behavioural and clinical factors contribute most strongly to predicted risk. This transparency makes the model more suitable for integration into clinical decision-support systems, especially in resource-limited settings.

Ethical use of patient data is another critical consideration. Although this study relies on anonymised, publicly available data, real-world deployment would involve sensitive personal and medical information. Ensuring data privacy, informed consent and compliance with data protection regulations is essential. Robust governance frameworks must be established to prevent misuse of patient data and to maintain public trust in ML-driven healthcare tools.

From a technical perspective, handling missing data and class imbalance remains a challenge. While imputation techniques helped preserve the dataset size, imputed values may introduce uncertainty and potentially affect predictions. Similarly, the relatively small number of positive biopsy cases makes it difficult to optimise sensitivity across all models. Addressing these issues through advanced imbalance-handling techniques or larger datasets would further improve model stability.

In summary, this study shows that machine learning models-particularly interpretable ones-can meaningfully assist cervical cancer risk assessment. However, careful attention to ethical considerations, population bias, transparency and clinical integration is essential to ensure that such systems are used safely, responsibly and effectively in real healthcare environments.

6. Conclusion

This project investigated the effectiveness of several machine learning algorithms in predicting cervical cancer biopsy outcomes using behavioural, demographic and clinical data. The study aimed to identify a model that could support early detection in settings where access to laboratory-based screening is limited. Because missing a positive case can have serious clinical consequences, the analysis prioritised **recall** as the most important performance metric *(Kessler, 2017)*.

Among all the models evaluated, **Logistic Regression (tuned)** achieved the highest recall, making it the most suitable model for early-stage screening applications. Although more complex models such as Random Forest and XGBoost demonstrated very high accuracy, they did not consistently outperform Logistic Regression in detecting positive biopsy cases. This finding aligns with previous research, which emphasises that simple, interpretable classifiers often perform strongly on behavioural datasets *(Varshini et al., 2024)*.

The use of systematic preprocessing - including missing-value imputation, feature scaling and stratified train–test splitting - helped ensure that modelling decisions reflected best practices in clinical ML research *(Fernandes, Cardoso & Fernandes, 2017)*. Feature importance analysis further validated the dataset by highlighting behavioural and clinical variables that correspond to known cervical cancer risk factors, such as STD history, smoking behaviour and reproductive characteristics *(Kessler, 2017)*.

Overall, this study demonstrates that machine learning can provide a practical and accessible tool for cervical cancer risk prediction, particularly in low-resource settings. A lightweight Logistic Regression model can be easily integrated into community health programmes, mobile health applications or decision-support systems to assist clinicians in identifying women at elevated risk.

Future work should explore methods to address class imbalance more effectively, expand the dataset with additional patient samples and test the models across broader populations. Incorporating more diverse data types - such as imaging or HPV genomic sequences - may further enhance predictive performance. Despite the study's limitations, the results contribute meaningful evidence that ML approaches can strengthen early detection efforts and improve women's health outcomes globally *(Varshini et al., 2024)*.

7. References

1. Fernandes, K., Cardoso, J.S. & Fernandes, J., 2017. *Transfer learning with partial observability applied to cervical cancer screening.* Iberian Conference on Pattern Recognition and Image Analysis (IBPRIA), pp.1–8. Available at: https://ieeexplore.ieee.org/abstract/document/10725056
2. Kessler, T.A., 2017. *Cervical cancer: Prevention and early detection.* Seminars in Oncology Nursing, 33(2), pp.172–183. Available at: https://www.sciencedirect.com/science/article/pii/S0749208117300153
3. Varshini, S.H., Aadhil, M., Sasvath, K.R., Thangamani, R. & Vimaladevi, M., 2024. *Cervical cancer prediction using machine learning.* 15th ICCCNT Conference, pp.1–7. Available at: https://archive.ics.uci.edu/dataset/383/cervical+cancer+risk+factors
4. Ribeiro, M.T., Singh, S. and Guestrin, C., 2016. *"Why should I trust you?": explaining the predictions of any classifier*. Proceedings of the 22nd ACM SIGKDD Conference, pp.1135–1144. Available at: https://doi.org/10.1145/2939672.2939778
5. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M. and Thrun, S., 2017. *Dermatologist-level classification of skin cancer with deep neural networks*. **Nature**, 542(7639), pp.115–118. Available at: https://doi.org/10.1038/nature21056
6. Obermeyer, Z. and Emanuel, E.J., 2016. *Predicting the future — big data, machine learning, and clinical medicine*. New England Journal of Medicine, 375(13), pp.1216–1219 Avialable at : https://doi.org/10.1056/NEJMp1606181
7. Arbyn, M., Weiderpass, E., Bruni, L., de Sanjosé, S., Saraiya, M., Ferlay, J. and Bray, F., 2020. *Estimates of incidence and mortality of cervical cancer in 2018*. The Lancet Global Health, 8(2), pp.e191–e203. Available at: https://doi.org/10.1016/S2214-109X(19)30482-6

8. Appendix

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier

Data=pd.read_excel("/content/cervical_cancer_DS.xlsx", na_values=['?'])

Data.info()

print(Data.head(5))

Data.isnull().sum()

missing_percent = Data.isnull().mean() * 100
print("\nPercentage of missing values per column:")
print(missing_percent)

Data.describe()

plt.figure(figsize=(12,6))
sns.heatmap(Data.isnull(), cbar=False, cmap='viridis')
plt.title("Missing Values Heatmap")
plt.show()

Data.hist(figsize=(15,10), bins=20, color='skyblue', edgecolor='black')
plt.suptitle("Feature Distributions", fontsize=10)
plt.tight_layout()
plt.subplots_adjust(top=0.9)
plt.show()

numeric_cols = Data.select_dtypes(include=['int64', 'float64']).columns
categorical_cols = Data.select_dtypes(include=['object']).columns

print("Numeric columns:", list(numeric_cols))
print("Categorical columns:", list(categorical_cols))

Mode_Age= Data['Age'].mode()[0]
print(Mode_Age)
```

```
Data['Age'].fillna(Mode_Age, inplace=True)

Mode_Sex_partners= Data['Number of sexual partners'].mode()[0]
print(Mode_Sex_partners)
Data['Number of sexual partners'].fillna(Mode_Sex_partners, inplace=True)

Data['First sexual intercourse'].fillna(Data['First sexual intercourse'].mean(),
inplace=True)

Data[numeric_cols[3]] = Data[numeric_cols[3]].fillna(2)

Data[numeric_cols[4]] = Data[numeric_cols[4]].fillna(0)

Data['Smokes (years)'].fillna(Data['Smokes (years)'].mean(), inplace=True)

Data['Smokes (packs/year)'].fillna(Data['Smokes (packs/year)'].mean(), inplace=True)

Data[numeric_cols[7]] = Data[numeric_cols[7]].fillna(1)

Data['Hormonal Contraceptives (years)'].fillna(Data['Hormonal Contraceptives
(years)'].mean(), inplace=True)

Data[numeric_cols[9]] = Data[numeric_cols[9]].fillna(0)

Data[numeric_cols[10:26]] = Data[numeric_cols[10:26]].fillna(0)

Data['STDs: Time since first diagnosis']=Data['STDs: Time since first diagnosis'].fillna(-
1)

Data['STDs: Time since last diagnosis']=Data['STDs: Time since last diagnosis'].fillna(-1)

Data.isnull().sum()

# spplittting target value
X = Data.drop('Biopsy', axis=1)
y = Data['Biopsy']

#converting into train 70% and test 30%
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

#Scaling Numerical Values
#it will be used for some models -- SVM, KNN, Logistic Regression
#Bcz These values are sensitive to the range of feature values
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled= scaler.transform(X_test)
```

```
#Linear Model
LG_Model= LogisticRegression(
   C=1.0,
   penalty="l2",
   solver="liblinear",
   max_iter=1000)
LG_Model.fit(X_train, y_train)
y_pred = LG_Model.predict(X_test)
print("Logistic Regression Results")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("LG - Confusion Matrix Heatmap")
plt.xlabel("Pred")
plt.ylabel("Actual")
plt.show()

LG_Model= LogisticRegression(max_iter=5000)
param_grid = {
   'penalty': ['l1', 'l2'],
   'C': [0.01, 0.1, 1, 10, 100],
   'solver': ['liblinear', 'saga']
}

grid = GridSearchCV(estimator=LG_Model, param_grid=param_grid, cv=5,
scoring='accuracy')
grid.fit(X_train_scaled, y_train)
print("Best Hyperparameters: ", grid.best_params_)
print("Best Score: ", grid.best_score_)
best_log_reg = grid.best_estimator_
y_pred = best_log_reg.predict(X_test_scaled)
print("Test Accuracy: ", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', linewidths=0.5)
plt.title("Confusion Matrix Heatmap - Logistic Regression")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

#Tree Based Model
rf_model = RandomForestClassifier(
```

```
    n_estimators=100,
    random_state=42,
    min_samples_split=5
    )
rf_model.fit(X_train, y_train)
y_pred = rf_model.predict(X_test)

print(" Model Results")
print("Accuracy:\n", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Greens')
plt.title("Random Forest ---- Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

#Keral Based Model
#svm uses svc for classification and svr for regression
svm_model = SVC(
    kernel='rbf',
    random_state=42,
    gamma="scale"
    )
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)

print(" SVM Results")
print("Accuracy:\n", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Oranges')
plt.title("SVM - Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

#Distance Based Model
knn_model = KNeighborsClassifier(
    n_neighbors=5,
    weights="distance",
    metric="euclidean"
    )

knn_model.fit(X_train, y_train)
y_pred = knn_model.predict(X_test)
```

```
print("K-Nearest Neighbors Results")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Purples')
plt.title("KNN -- Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

from xgboost import XGBClassifier

xgb = XGBClassifier(
   use_label_encoder=False,
   eval_metric='logloss',
   random_state=42)

xgb.fit(X_train, y_train)
y_pred = xgb.predict(X_test)

print("XGBoost Results ")
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))

sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='coolwarm')
plt.title("XGBoost - Confusion Matrix")
plt.show()

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam


# Model Building
model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compile model
model.compile(
   optimizer=Adam(learning_rate=0.0005),
   loss='binary_crossentropy',
   metrics=['accuracy']
   )
```

```
#Model Training
history = model.fit(X_train, y_train, epochs=50, batch_size=16, validation_split=0.2,
verbose=0)

# Evaluating
loss, acc = model.evaluate(X_test, y_test, verbose=0)
print("Neural Network Results")
print("Accuracy:", acc)

# Predictions (for confusion matrix)
y_pred = (model.predict(X_test) > 0.5).astype("int32")

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("Neural Network - Confusion Matrix Heatmap")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

plt.figure(figsize=(12,5))

# Loss Curve
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title("Training vs Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.ylim(0, 1)
plt.legend()
#Accuracy curve
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title("Training vs Validation Accuracy")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.ylim(0, 1)
plt.legend()

plt.tight_layout()
plt.show()

Data1=Data

Data1.head()
```

```
Data1 = Data1.drop(['STDs: Time since first diagnosis', 'STDs: Time since last
diagnosis'], axis=1)

Data1.isnull().sum()

X1 = Data1.drop('Biopsy', axis=1)
y1 = Data1['Biopsy']

X_train1, X_test1, y_train1, y_test1 = train_test_split(X1, y1, test_size=0.3,
random_state=42)

scaler = StandardScaler()
X_train_scaler1 = scaler.fit_transform(X_train1)

X_test_scaler1 = scaler.transform(X_test1)

rf_model = RandomForestClassifier(
    n_estimators=100,
    random_state=42,
    min_samples_split=5)
rf_model.fit(X_train1, y_train1)
y_pred = rf_model.predict(X_test1)

print(" Model Results")
print("Accuracy:\n", accuracy_score(y_test1, y_pred))
print("Classification Report:\n", classification_report(y_test1, y_pred))

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test1, y_pred), annot=True, fmt='d', cmap='Greens')
plt.title("Random Forest ---- Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

svm_model = SVC(
    kernel='rbf',
    random_state=42,
    gamma="scale"
    )
svm_model.fit(X_train1, y_train1)
y_pred = svm_model.predict(X_test1)

print(" SVM Results")
print("Accuracy:\n", accuracy_score(y_test1, y_pred))
print("Classification Report:\n", classification_report(y_test1, y_pred))

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test1, y_pred), annot=True, fmt='d', cmap='Oranges')
```

```
plt.title("SVM - Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

knn_model = KNeighborsClassifier(
   n_neighbors=5,
   weights="distance",
   metric="euclidean"
   )
knn_model.fit(X_train1, y_train1)
y_pred = knn_model.predict(X_test1)

print("K-Nearest Neighbors Results")
print("Accuracy:", accuracy_score(y_test1, y_pred))
print("\nClassification Report:\n", classification_report(y_test1, y_pred))

plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test1, y_pred), annot=True, fmt='d', cmap='Purples')
plt.title("KNN -- Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

from xgboost import XGBClassifier

xgb = XGBClassifier(
   use_label_encoder=False,
   eval_metric='logloss',
   random_state=42)
xgb.fit(X_train1, y_train1)
y_pred = xgb.predict(X_test1)

print(" XGBoost Results ")
print("Accuracy:", accuracy_score(y_test1, y_pred))
print("\nClassification Report:\n", classification_report(y_test1, y_pred))

sns.heatmap(confusion_matrix(y_test1, y_pred), annot=True, fmt='d', cmap='coolwarm')
plt.title("XGBoost - Confusion Matrix")
plt.show()

LG_Model= LogisticRegression(max_iter=2000)
LG_Model.fit(X_train1, y_train1)
y_pred = LG_Model.predict(X_test1)
print("Logistic Regression Results")
print("Accuracy:", accuracy_score(y_test1, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test1, y_pred))
print("Classification Report:\n", classification_report(y_test1, y_pred))
```

```
plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test1, y_pred), annot=True, fmt='d', cmap='Blues')
plt.title("LG - Confusion Matrix Heatmap")
plt.xlabel("Pred")
plt.ylabel("Actual")
plt.show()

LG_Model= LogisticRegression(max_iter=5000)
param_grid = {
    'penalty': ['l1', 'l2'],
    'C': [0.01, 0.1, 1, 10, 100],
    'solver': ['liblinear', 'saga']
}

grid = GridSearchCV(estimator=LG_Model, param_grid=param_grid, cv=5,
scoring='accuracy')
grid.fit(X_train_scaler1, y_train1)
print("Best Hyperparameters: ", grid.best_params_)
print("Best Score: ", grid.best_score_)
best_log_reg = grid.best_estimator_
y_pred = best_log_reg.predict(X_test_scaler1)
print("Test Accuracy: ", accuracy_score(y_test1, y_pred))
print(classification_report(y_test1, y_pred))

cm = confusion_matrix(y_test1, y_pred)

plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', linewidths=0.5)
plt.title("Confusion Matrix Heatmap - Logistic Regression")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

y_pred = best_log_reg.predict(X_test_scaler1)

print("Predictions:", y_pred[:30])
print("Actual:", y_test.values[:30])

models = [
    "Logistic Regression",
    "Random Forest",
    "XGBoost",
    "SVM",
    "KNN",
    "ANN"
]

# Values from your table
```

```
accuracy = [96.12, 95.73, 94.57, 94.18, 93.79, 95.34]
recall = [60, 47, 47, 0, 0, 47]

x = np.arange(len(models))
width = 0.35

plt.figure(figsize=(12, 6))

bars1 = plt.bar(x - width/2, accuracy, width, label="Accuracy")
bars2 = plt.bar(x + width/2, recall, width, label="Recall")

# Add value labels on bars
for bar in bars1:
    height = bar.get_height()
    plt.text(
        bar.get_x() + bar.get_width()/2,
        height + 0.5,
        f"{height:.2f}",
        ha="center",
        va="bottom",
        fontsize=9
    )

for bar in bars2:
    height = bar.get_height()
    plt.text(
        bar.get_x() + bar.get_width()/2,
        height + 0.5,
        f"{height:.0f}",
        ha="center",
        va="bottom",
        fontsize=9
    )

# Labels and title
plt.xlabel("Models")
plt.ylabel("Percentage (%)")
plt.title("Model Comparison: Accuracy vs Recall")
plt.xticks(x, models, rotation=30)
plt.legend()

plt.tight_layout()
plt.show()

log_reg = LogisticRegression(
    C=1.0,
    penalty="l2",
    solver="liblinear",
```

```
    max_iter=500
)

log_reg.fit(X_train, y_train)
coefficients = log_reg.coef_[0]

# Feature names
feature_names = X.columns

coef_df = pd.DataFrame({
    "Feature": feature_names,
    "Coefficient": coefficients
})

coef_df["Abs_Coefficient"] = coef_df["Coefficient"].abs()
coef_df = coef_df.sort_values(by="Abs_Coefficient", ascending=False)

# Plot
plt.figure(figsize=(12, 6))
plt.bar(coef_df["Feature"], coef_df["Coefficient"])
plt.axhline(0, linestyle="--")
plt.xticks(rotation=90)
plt.xlabel("Features")
plt.ylabel("Coefficient Value")
plt.title("Feature Importance (Logistic Regression)")
plt.tight_layout()
plt.show()

new_patient_df = pd.DataFrame(
    np.zeros((1, len(scaler.feature_names_in_))),
    columns=scaler.feature_names_in_
)

# some known values
new_patient_df.loc[0, "Age"] = 23
new_patient_df.loc[0, "Number of sexual partners"] = 3
new_patient_df.loc[0, "First sexual intercourse"] = 18
new_patient_df.loc[0, "Num of pregnancies"] = 3
new_patient_df.loc[0, "Smokes"] = 1
new_patient_df.loc[0, "Smokes (years)"] = 0
new_patient_df.loc[0, "Smokes (packs/year)"] = 0
new_patient_df.loc[0, "Hormonal Contraceptives"] = 1
new_patient_df.loc[0, "Hormonal Contraceptives (years)"] = 5
new_patient_df.loc[0, "IUD"] = 1
new_patient_df.loc[0, "IUD (years)"] = 5
new_patient_df.loc[0, "STDs"] = 1
new_patient_df.loc[0, "STDs (number)"] = 4
new_patient_df.loc[0, "STDs:condylomatosis"] = 0
```

```
new_patient_df.loc[0, "STDs:cervical condylomatosis"] = 1
new_patient_df.loc[0, "STDs:vaginal condylomatosis"] = 1
new_patient_df.loc[0, "STDs:vulvo-perineal condylomatosis"] = 1
new_patient_df.loc[0, "STDs:syphilis"] = 0
new_patient_df.loc[0, "STDs:pelvic inflammatory disease"] = 1
new_patient_df.loc[0, "STDs:genital herpes"] = 1
new_patient_df.loc[0, "STDs:molluscum contagiosum"] = 1
new_patient_df.loc[0, "STDs:AIDS"] = 1
new_patient_df.loc[0, "STDs:HIV"] = 0
new_patient_df.loc[0, "STDs:Hepatitis B"] = 1
new_patient_df.loc[0, "STDs:HPV"] = 1
new_patient_df.loc[0, "STDs: Number of diagnosis"] = 5
new_patient_df.loc[0, "Dx:Cancer"] = 1
new_patient_df.loc[0, "Dx:CIN"] = 1
new_patient_df.loc[0, "Dx:HPV"] = 1
new_patient_df.loc[0, "Dx"] = 1
new_patient_df.loc[0, "Hinselmann"] = 1
new_patient_df.loc[0, "Schiller"] = 0
new_patient_df.loc[0, "Citology"] = 1

# All other features remain 0

new_patient_scaled = scaler.transform(new_patient_df)

prediction = best_log_reg.predict(new_patient_scaled)
probability = best_log_reg.predict_proba(new_patient_scaled)

print("Prediction:", "Cancer Detected" if prediction[0] == 1 else "No Cancer Detected")
print("Cancer Probability:", round(probability[0][1] * 100, 2), "%")

#This explains why the model predicted this result.
coeffs = best_log_reg.coef_[0]
contributions = new_patient_scaled[0] * coeffs

contrib_df = pd.DataFrame({
    "Feature": scaler.feature_names_in_,
    "Contribution": contributions
})

contrib_df["Abs_Contribution"] = contrib_df["Contribution"].abs()
contrib_df = contrib_df.sort_values(by="Abs_Contribution", ascending=False)

print(contrib_df.head(10))
```