Cervical Cancer Prediction Using XG boost

TR Vedhavathy[1], Gnana Ganesh[2]
Department of Networking and Communication, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Kattankalathur - 603203, Chennai, Tamilnadu, India
gg8666@srmist.edu.in, vedhavat@srmist.edu.in

***Abstract* – Cervical cancer is a disease that predominantly affects women and is frequently fatal. Contrarily, early cervical cancer identification reduces mortality and other impacts.**

Early detection of cervical cancer might be aided by risk factors. We suggested research for the early diagnosis of cervical cancer using three ensemble-based classification approaches, including extreme Gradient Boosting (XGBoost) for optimisation. The study makes use of the 36 risk variables and the four objectives from the data collection on cervical cancer risk factors (Hinselmann, Schiller, Cytology, and Biopsy). The four objectives form the cornerstone of the most widely used cervical cancer detection techniques. The success of the suggested research is evaluated using its accuracy, sensitivity, specificity, positive predictive accuracy (PPA), and negative predictive accuracy (NPA).

A technique was also used to provide better results with less characteristics. Less traits have also resulted in better outcomes. When compared to earlier benchmark studies for cervical cancer detection using a reduced risk factors data set, the suggested models exhibit noticeably better accuracy.

I. Introduction

1.1Understanding the XGBoost Algorithm

The XGBoost technique is beneficial since it can be used for both classification and regression applications. It utilises supervised learning and gradient boosting to create a multi-decision tree prediction model.

The algorithm works by improving its future projections by learning from previous model predictions. It creates weak models progressively. It generates a preliminary model based on the training data, and the residuals—or first model errors—are utilised to generate a second model. This approach is repeated until the maximum number of models have been developed or the model consistently provides correct predictions.

II. Method

2.1 Importing the Libraries and Datasets:

Importing our libraries and information onto our environment is the initial step in creating our model.

2.1.1 Libraries listed:

Pandas:

The most well-liked Python data analysis and manipulation package is *pandas*. It is mostly helpful for manipulating dataframes in this project.

NumPy:
Large, multi-dimensional arrays and matrices may be worked with and manipulated more easily using high-level mathematical methods provided by the NumPy Python library.

Joblib:

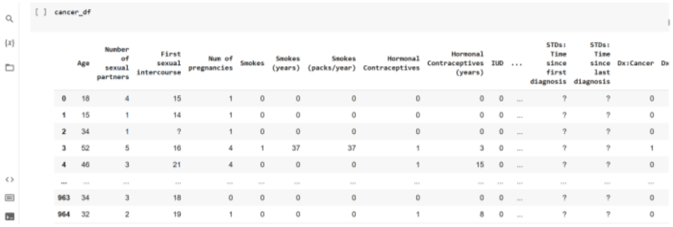
The Python Joblib package enables the parallel execution of computationally expensive activities. It provides a set of tools for managing massive amounts of data simultaneously and retaining the outcomes of computationally difficult activities. Joblib is highly useful for machine learning models since it allows you to preserve the state of your computation and start working later or on another computer.

Data visualisation tools include *seaborn* and *matplotlib.pyplot*. *plotly* utilised for interactive data visualisation

To import the data we gathered, we may utilise pandas after configuring our libraries. The data set, which contains information on 900 instances, must be stored in a CSV file.

2.2 Exploratory Data Analysis

Performing a qualitative analysis of the data is essential to eliminate errors from the database; it necessitates deleting duplicates, fixing mistakes, and dealing with missing numbers.



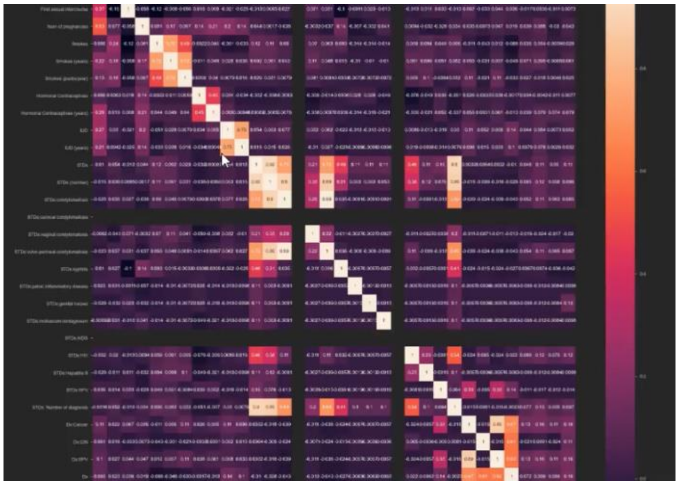
```
[ ] cancer_df
```

	Age	Number of sexual partners	First sexual intercourse	Num of pregnancies	Smokes	Smokes (years)	Smokes (packs/year)	Hormonal Contraceptives	Hormonal Contraceptives (years)	IUD	...	STDs: Time since first diagnosis	STDs: Time since last diagnosis	Dx:Cancer	Dx
0	18	4	15	1	0	0	0	0	0	0	...	?	?	0	
1	15	1	14	1	0	0	0	0	0	0	...	?	?	0	
2	34	1	?	1	0	0	0	0	0	0	...	?	?	0	
3	52	5	16	4	1	37	37	1	3	0	...	?	?	1	
4	46	3	21	4	0	0	0	1	15	0	...	?	?	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
963	34	3	18	0	0	0	0	0	0	0	...	?	?	0	
964	32	2	19	1	0	0	0	1	8	0	...	?	?	0	

2.3 Data Visualization

In order to find the connections between the characteristics, we will help visualise the data.
The first stage is to build your network of connections:

Annot = True displays the numeric values within the heatmap, whereas (30, 30) makes a much bigger heatmap so that all data is displayed and fits on the heatmap. When drawn, it appears as follows:



The effect or link between each attribute in the dataset is shown in this matrix. Nearly perfect correlation is shown by values close to 1 (white), virtually no association is shown by values close to 0 (purples), and inverse correlation is shown by negative values (darkest). As you can see, identical features line up on the x and y axes and exhibit a perfect correlation with one another, resulting in a diagonal strip of white values that seems to have a perfect correlation.

2.4 Preparing Data Model Training

Before training the model, we must first prepare our data.

The remaining four features inside the data, as previously described, are **"Hinselman," "Schiller," "Citology,"** and **"Biopsy".** All four of these characteristics are diagnostic measures. However, because we are only attempting to train the algorithm to predict the target variable biopsy, we have to regulate our goal produce to that column.

The last step before our data is ready to be given to the model is to scale it. To do this, we must import the Python package scikit-learn, which includes a variety of classification, regression, and clustering techniques.

To use SciKitLearn to normalise our input data or our features, we need to import the StandardScaler and MinMaxScaler.

We also want to make a new instance of the "scaler" object from our class, "StandardScaler." We next wish to apply the "fit_transform" function on our object, "scaler," and send it on to X to obtain an output that is a scaled version of X.

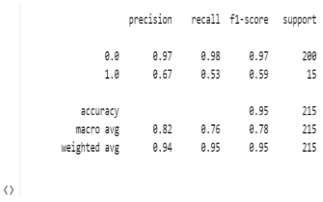
2.4 Training and Evaluating the Model

Installing xgboost is required before we can train and test the model. To do this, run the following code:

install xgboost with pip
Following that, we wish to import xgboost as xgb. We may then train the xgboost classification model after doing this.

As 0.1 is probably a decent starting point for our system, we can set our learning rate at that value. The longest route from the decision tree's base to the root is 5, therefore we may choose it as the required depth for our classification tree. Here, we may additionally define the number of models or estimators that we'll use. The number of estimators can be adjusted to 10.
It is crucial to remember that we may significantly expand the depth of the tree and the number of estimators. Our model, however, is growing increasingly sophisticated as a result of this. Overfitting of the training data is a potential issue. The

model could perform admirably on the training set of data while failing miserably on the testing set,



```
              precision    recall  f1-score   support

         0.0       0.97      0.98      0.97       200
         1.0       0.67      0.53      0.59        15

    accuracy                           0.95       215
   macro avg       0.82      0.76      0.78       215
weighted avg       0.94      0.95      0.95       215
```

III. Result

User-provided data are fed into the algorithm. Age, STDs, and other information will be entered. Regression and classification are used to determine whether a biopsy should be done or not. Consequently, the project aids in our self-analysis.

The algorithm receives the data that users supply. Information on age, STDs, and other metrics will be entered. The use of regression and classification to decide whether or not to perform a biopsy. All input factors age,STDs, other metrics will undergo regression and after that, classification for biopsy. The initiative therefore supports our self-analysis.

IV. Conclusion

This research analyses several ensemble Gradient Boosting algorithms for cervical cancer detection. The data set, which contains 858 records and 36 features, was obtained from the UCI machine learning library. One of the variables under consideration is the cervical cancer diagnostic test. For each target class, separate trials were carried out. The firefly approach was used in data preprocessing to uncover the essential features and refine the models. The tests were done with 30 attributes to compare the performance of the models. Boosting a relatively steep grade For subsequent work, the model will be validated using multiple data sets. It is also vital to investigate alternate models that can manage outliers and imbalanced data in various ways.

Acknowledgment

I would want to express my sincere gratitude to each and every one of them. I am very appreciative of Dr. T. R. Vedhavathy's help with the research, especially her direction, frequent oversight, and provision of necessary details.

References

1. F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries", *CA.*
2. T. S. Shylasree et al., "Quality of life in long term survivors of cervical cancer: A cross sectional study", *Indian J.*
3. B. Wang, X. Lv, M. I. N. Li and J. Wang, "Classification of Cervical Biopsy Images Based on LASSO and EL-SVM",
4. J. Li et al., "XGBoost Classifier Based on Computed Tomography Radiomics for Prediction of Tumor-In filtrating CD8 + T-Cells in Patients With Pancreatic Ductal Adenocarcinoma"