AI-driven intelligent forecasting of medical conditions with random forest classifier and Naive Bayes

N.vijay Kumar [1], Boddu Avinash[2], R.Gopichandh[3], R.SaiPhanidra[4]
[1] Professor, [2, 3 & 4] Student
[1]nvk20022001@gmail.com, [2] avinashboddu3@gmail.com,, [3] rajarapugopichand9@gmail.com, [4] phanisai@gmail.com
Department of Computer Science and Engineering,
Narasaraopeta Engineering College, Narasaraopet, Andhra Pradesh, India

***ABSTRACT* - In the healthcare field, a wealth of patient data, ranging from reported symptoms to detailed lab results, is routinely collected. This extensive dataset is crucial for physicians as they strive to accurately diagnose various medical conditions. However, with the integration of artificial intelligence (AI) techniques, new opportunities for disease classification have emerged. Machine learning algorithms like Naive Bayes now play a key role in this area. By leveraging AI, these algorithms improve traditional diagnostic methods, leading to more effective and precise disease identification.**

In this study, our main goal is to use these methodologies to enhance the performance of disease classification algorithms. Through careful evaluation, we aim to determine how effective Naive Bayes and Random Forest algorithms are when applied to disease datasets. By comparing their results to those of conventional diagnostic methods, we hope to highlight the potential of AI-driven solutions in advancing disease diagnosis and management. In terms of using AI[1] to enhance healthcare procedures, thisresearch constitutes a substantial advancement.

KEYWORDS*:*Healthcare Patient Data Disease Diagnosis Artificial Intelligence (AI) Naive Bayes Random Forest Naive Bayes Machine Learning Algorithms Preprocessing Techniques SMOTE Outlier Removal Performance Analysis Disease Classification Diagnostic Paradigms Medical Expertise Enhanced Diagnosis Management Improvement

I.INTRODUCTION

Modern lifestyle [2] and dietary habits contribute to the increased prevalence of diabetes, heart disease, and cancer. Factors such as lack of regular exercise, consumption of processed foods and sugary drinks, smoking, and chronic stress contribute to the rising incidence of these health conditions. Diabetes, characterized by high blood sugar levels, is strongly linked to unhealthy lifestyle choices, including poor diet and physical inactivity. Similarly, heart disease, the leading cause of death globally, often stems from a combination of factors such as an unhealthy diet, lack of exercise, smoking, and obesity. Cancer, another significant health concern, can be influenced by dietary factors such as consumption of processed meats and sugary drinks, along with other lifestyle choices. By aiding in disease diagnosis, maximizing treatment plans, identifying fraud, and forecasting future health hazards, data mining [3] is essential in tackling these health issues.

By harnessing AI algorithms, healthcare systems can glean insights from vast datasets, discern intricate patterns, and offer precise predictions regarding disease progression, treatment effectiveness, and potential risk factors. This enables more effective management and prevention of conditions such as diabetes, heart disease, and cancer, ultimately improving patient outcomes and quality of life.

Our main goal is to create predictive models that, when combined with classification algorithms, can correctly identify diseases within a given dataset. In particular, we have used diabetes, coronary heart disease, and cancer as three different illness datasets to train the classifiers like Random Forest and Naïve Bayes .The Random Forest algorithm was selected to tackle the problem of overfitting that is frequently linked to individual decision tree models. In comparison to the Naive Bayes method, Random Forest reduces overfitting and improves prediction accuracy by combining the results of several decision trees.We performed thorough[4] analyses to evaluate the performance of each model, computing a range of performance metrics including, accuracy, sensitivity, F1 score , specificity and area under the ROC curve. These measures enable us to evaluate how well each classifier performs in correctly recognizing disease.

The most commitment of this paper can be summarized as takes:

- Dataset Selection: The research makes use of disease datasets that include cancer, heart disease, and diabetes because these conditions are known to pose serious risks to people's health. These databases include generic disease statistics as well as vital information about patient healthcare.

- Data Preprocessing: The input datasets go through preprocessing procedures to prepare the data for additional analysis before analysis begins. This include managing missing values, locating and eliminating outliers, and evaluating the relationships between different variables. Furthermore, the Smote is utilized to mitigate class imbalance and guarantee the classification process's resilience.

- Data Splitting: Thirty percent of the preprocessed datasets are set aside for testing, while the remaining seventy percent are used for training. This division preserves data integrity while guaranteeing an effective analysis.

- Data Mining Algorithms:
  To evaluate the system's effectiveness with respect to the provided disease datasets, the study uses algorithms in data mining, like Gaussian Naïve Bayes and Random Forest . These algorithms are selected based on how well they perform in classification tasks and how well they work with medical data analysis. Furthermore, a comparison of the acquired categorization results with previous research shows significant gains in predicting accuracy.

II. LITERATURE SURVEY

Jakka et al. [5] employed an AI-based approach for disease prediction utilizing classifiers like bayes and random forest. Their objective was to determine which classifiers could predict diabetes in patients with the highest accuracy and precision. The study utilized six classification algorithms, on the Pima Indians diabetes dataset containing nine features. It was found Among the six classifiers tested, logistic regression achieved the highest accuracy of 77.6%.

The adoption of unique machine learning techniques for better diabetes prediction was explained by Karun et al. [6]. Each show's execution is calculated both with and without the inclusion of extraction preparation. The diabetes dataset with 8 quality and 1 predicted course has been used for the investigation. Their most recent research found that while calculated with some algorithms appear superior expectation exactness without highlight extraction, back vector machine, K-means closest neighbor (KNN) and few algorithms appear superior expectation exactness with include extraction .Leiherer et al. promoted more information on betatrophin organization and control to estimate cardiac fatality in coronary patients[7].

Wu et al.'s [8] cardiovascular disease prediction framework integrates multiple approaches into a unified protocol known as hybridization. This approach combines various methods to yield a more accurate diagnosis. The primary objective of their research is to propose a novel machine learning methodology aimed at improving the accuracy of predicting cardiac diseases.

Baad et al. [9] have out an analysis of the various techniques used by the researchers to determine the heart disease based on a patient's real information.Chala Beyene et al. (2018) used one or two machine computations to predict the occurrence of cardiovascular illness using artificial neural networks, Naïve Bayes, choice plants, give assistance point producers, and k-nearest neighbor [10]. In addition to Bayesian framework subgraphs (root hubs to arise from events), which are used as the details from which PAI-2-associated affect handle was obtained, Corsetti et al. [11] presented Bayesian alternatives in treatment.To obtain precision, the Smote, correlation coefficient application, and ten times cross-validations are coupled to the breast cancer data [12].

Sireesha Moturi et al. [13] utilized a data pickup method as an inclusive selection procedure to streamline the search process and focus on relevant data. This method aims to reduce the complexity of the dataset by selecting only the most informative features, thereby improving the efficiency of subsequent analysis. The effectiveness of this strategy was evaluated using medical data from the NRI Healing Center, where it was tested and validated..Amrane et al. [14] developed a model utilizing Credulous Bayes and K-nearest neighbor methods. They achieved accuracy rates of 96.19% and 97.51%, respectively.Shahidi et al. [15] observed variations in precision scores across different machine learning models. This suggests that additional factors such as excluding exceptions, filling in missing values, and utilizing inclusion selection methods can impact the models' ability to achieve the highest levels of precision.

III. PROPOSED METHODOLOGY

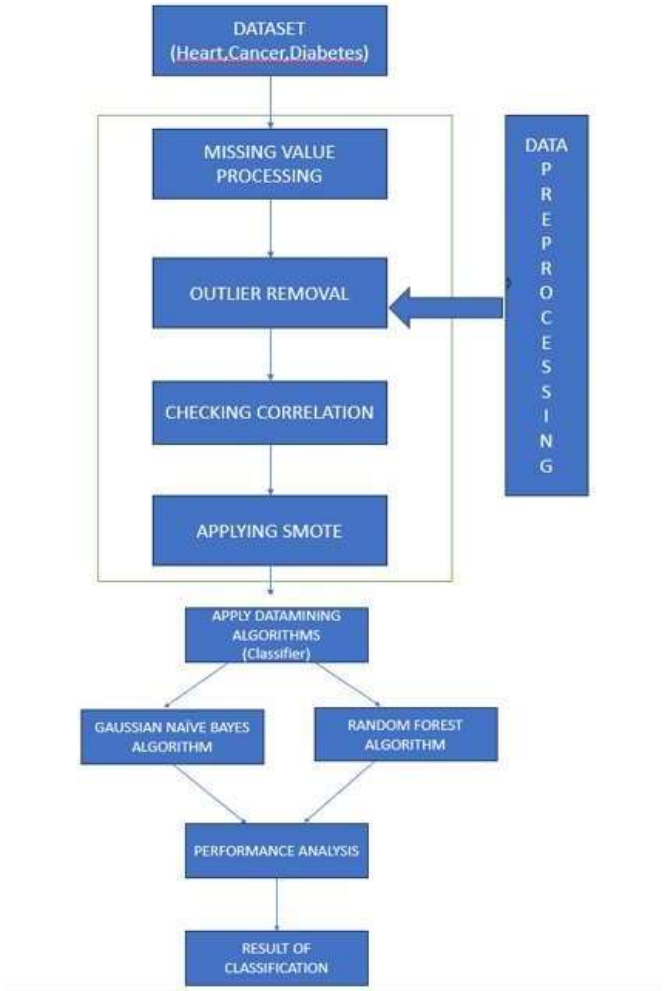

Fig-1:proposed methodology

The methodology proposed in this research leverages Anaconda for conducting thorough data analysis, utilizing its comprehensive package management system designed for predictive analysis and data manipulation. In this study, three specific patient datasets concerning diabetes, coronary heart disease, and breast cancer were selected due to their significant impact and suitability for comparative analysis based on shared characteristics, as depicted in Figure 1.

Initially, the datasets underwent loading and thorough inspection for any missing values, which were subsequently substituted with null values upon detection. Following this, an examination of the inter-column correlations within the datasets was conducted, leading to the removal of any correlated columns to enhance dataset efficiency. Additionally, boolean values were standardized to 1 and 0 to ensure uniformity across the datasets.

The original datasets were then partitioned into training and testing subsets, with 70% of the data is allocated for training purposes and the remaining 30% used for testing. To estimate the performance of the Naive Bayes algorithm, the training data was utilized, and accuracy metrics were computed for each disease class through the utilization of confusion matrices.

Likewise, the random forest algorithm is applied to the training data and calculated accuracy metrics for each disease class individually. The model's internal parameters were updated through successive epochs until errors in the datasets were minimized.To further estimate the method's effectiveness sample test data is applied for each disease class to train models separately to ascertain their ability to accurately identify the presence of the disease. A comparison of results between the Naive Bayes and random forest models indicated that the latter provided more precise classifications.

By incorporating advanced preprocessing techniques such as outlier removal and Smote, we further improved the method's performance. Additionally, integrating ROC curves and AUC scores allowed for a comprehensive evaluation of model performance, providing a deeper understanding of classification accuracy.

The proposed method showcases potential applicability for real-time disease data classification, facilitating healthcare professionals in promptly identifying whether a patient is affected by a specific disease. Through the integration of advanced preprocessing techniques and comprehensive evaluation metrics, the method presents a robust framework for disease diagnosis and management in clinical settings.

IV. RESULTS AND DISCUSSION

IV. DATASET

The study's dataset includes several disease datasets, such as those related to diabetes, heart disease, and breast cancer. These datasets, which offer insightful information about a range of medical disorders, were gathered utilizing wearable technology and predictive data.

A. Diabetes Dataset

The diabetes dataset is taken from the National Institute of Diabetes Digestive and Kidney Diseases (NIDDK). Data from female Pima Indian individuals who are at least 21 years old is included. This dataset contains the following attributes in Fig-2:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Fig-2 Diabetes dataset

B. Heart Disease Dataset

The Framingham Heart Study provided the dataset for coronary heart disease, which covers a number of risk factors related to medical history and demographics. This dataset contains the following attributes in Fig-3:

```
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   male             4240 non-null   int64
 1   age              4240 non-null   int64
 2   education        4135 non-null   float64
 3   currentSmoker    4240 non-null   int64
 4   cigsPerDay       4211 non-null   float64
 5   BPMeds           4187 non-null   float64
 6   prevalentStroke  4240 non-null   int64
 7   prevalentHyp     4240 non-null   int64
 8   diabetes         4240 non-null   int64
 9   totChol          4190 non-null   float64
 10  sysBP            4240 non-null   float64
 11  diaBP            4240 non-null   float64
 12  BMI              4221 non-null   float64
 13  heartRate        4239 non-null   float64
 14  glucose          3852 non-null   float64
 15  TenYearCHD       4240 non-null   int64
dtypes: float64(9), int64(7)
memory usage: 530.1 KB
```

Fig-3 Heart dataset

C.Breast Cancer Dataset

The breast cancer dataset is sourced from the Breast Cancer Wisconsin dataset. It includes various attributes related to breast cancer diagnosis. The attributes present in this Fig-4:

```
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 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
 32  Unnamed: 32              0 non-null      float64
dtypes: float64(31), int64(1), object(1)
memory usage: 146.8+ KB
```

Fig-4 breast cancer dataset

These datasets provide comprehensive information essential for training and evaluating predictive models for disease diagnosis. The attributes encapsulate diverse aspects of each medical condition, enabling robust analysis and classification. Through the application of machine learning algorithms and rigorous evaluation, we aim to enhance disease diagnosis and management practices

DATA VISUALIZATION

The distribution of the outcome variable in the data was examined and visualized.

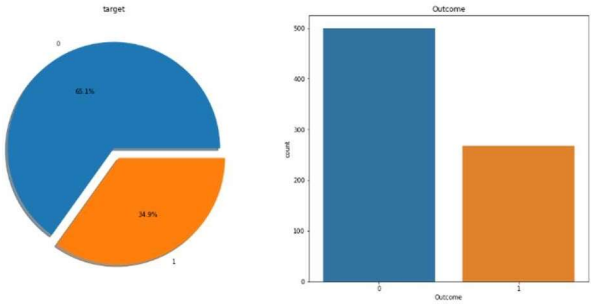

Fig-5 outcome variable visualized

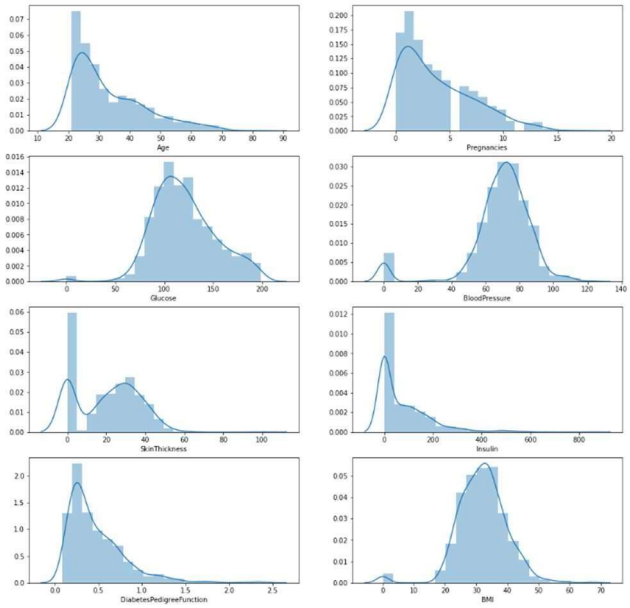

Fig-6 Subplots displaying the distribution of various features

Subplots Fig-6 displaying the distribution of various features from the DataFrame df, including various columns. Each subplot represents the distribution of a specific feature using a kernel density estimate plot with 20 bins. This visualization helps in understanding the distribution and spread of each feature within the dataset, aiding in exploratory data analysis and identifying potential patterns or outliers.

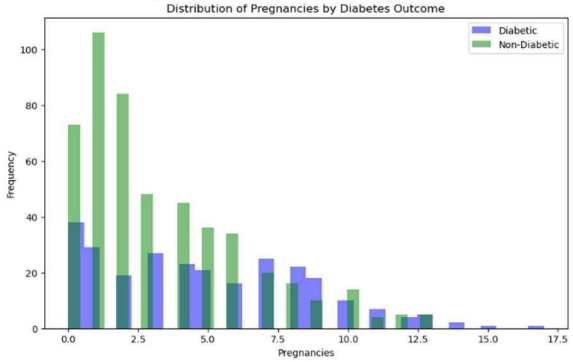

Fig-7 Pregnancies vs outcome

The graph Fig-7 depicts the distribution of pregnancies among diabetic and non-diabetic individuals. The x-axis represents the number of pregnancies, while the y-axis indicates the frequency of occurrences. Diabetic pregnancies show a higher frequency at lower values on the x-axis, implying poorer outcomes, compared to non-diabetic pregnancies, which exhibit higher frequencies at higher values on the x-axis.

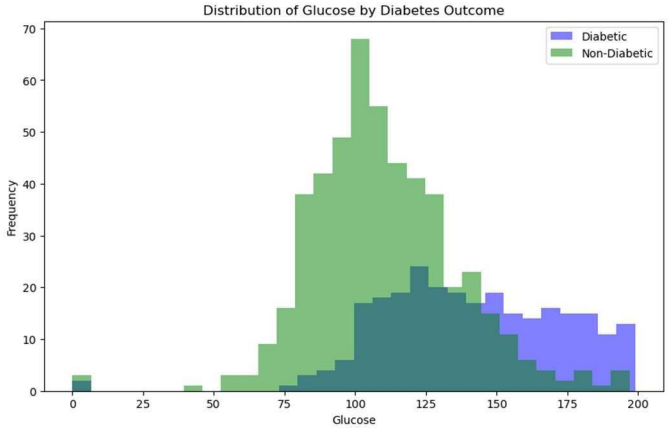

Fig-8 Glucose vs Outcome

The graph Fig-8 illustrates the distribution of blood glucose levels categorized by diabetes outcome. On the x-axis is the blood glucose levels, while the y-axis represents frequency. Diabetic individuals exhibit a higher frequency of blood glucose levels between 100 to 175 mg/dL, whereas non-diabetic individuals display lower frequencies across the entire range, peaking between 75 to 125 mg/dL.

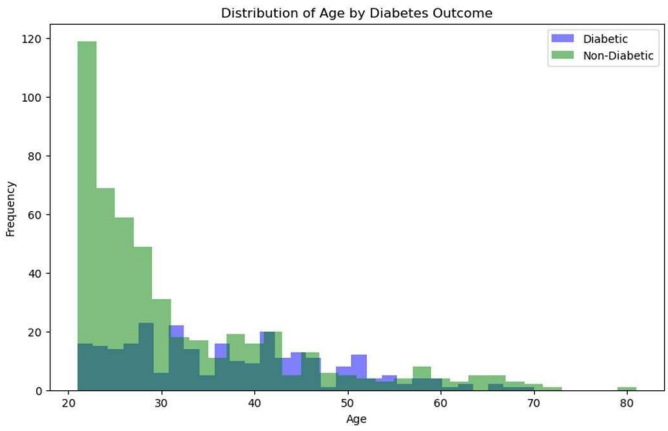

Fig-9 Age vs Outcome

The Fi-9 titled "Distribution of Age by Diabetes Outcome" depicts a bar graph with age plotted on the x-axis and frequency on the y-axis. It showcases two bars representing "Diabetic" and "Non-Diabetic" groups. The data illustrates a correlation between age and diabetes prevalence, with the highest frequency of diabetic individuals observed in the 60-70 age range compared to other age groups depicted in the graph.

V.PREPROCESSING TECHNIQUES

V.I Filling Missing Values

Since missing data can negatively impact machine learning model performance, filling in the gaps in data is an essential step in preparing data for analysis. Various methods exist for managing absent values [16], contingent upon the characteristics of the information and the particular demands of the examination. The following are a few typical methods for adding missing values:

1.Mean/Median/Mode Imputation: Use the corresponding feature's mean, median, or mode to fill in any missing values. This method preserves the overall distribution of the data and works well with numerical data.

2.Forward Fill/Backward Fill: To fill in missing values in time series or sequential data, propagate the most recent known value forward or the subsequent known value backward.

The features of the dataset and the possible effects of each imputation technique on the analysis outcomes must be carefully taken into account. To further guarantee the accuracy of the imputed data, it is advised to assess the effectiveness of the selected imputation methodology using cross-validation or other validation techniques.

V.II. Outliers Removal

In order to prevent abnormal data points from having an excessive impact on the analysis or modeling process, removing outliers from a dataset is an essential step in the data preprocessing process. Measurement errors, data entry errors, and real unusual events are just a few of the causes of outliers. The following are typical methods for eliminating outliers:
A simple method for determining outliers based on their departure from the dataset mean is the standard deviation method. This is how it operates:
Calculate the Mean and Standard Deviation: Compute the mean (average) and standard deviation of the dataset.

1. Mean (μ) = Σxi / n Standard Deviation (σ) = √(Σ(xi - μ)² / n)

   Where:

   - xi represents each data point in the dataset.
   - n is the total number of data points

2. *Identify Outliers*: Determine the threshold for identifying outliers. In the Standard Deviation Method, outliers are typically defined as data points that fall outside of ±3 standard deviations from the mean.

   Upper Threshold = μ + 3σ Lower Threshold = μ - 3σ

   Any data points that exceed the upper threshold or fall below the lower threshold are considered outliers.

3. *Remove Outliers:* Once the outliers are identified, they can be removed from the dataset.
   - For numerical datasets, outliers can be replaced with NaN (Not a Number) or deleted entirely.
   - For categorical datasets, outliers may be removed if they represent erroneous or rare categories.
4. *Reevaluate the Dataset*: After removing outliers, it's essential to reassess the dataset's distribution and statistical properties to ensure that the removal process did not unduly affect the dataset's integrity or bias the analysis results.

*Optional:Adjust Threshold:*Depending on the dataset's characteristics and specific requirements of the analysis, the threshold for identifying outliers (e.g., ±3 standard deviations) can be adjusted to be more or less stringent.

The Standard Deviation Method provides a simple and intuitive way to identify outliers based on their deviation from the mean of the dataset. However, it's important to exercise caution when using this method, as outliers[17,18] may sometimes contain valuable information or represent genuine data points that should not be discarded indiscriminately.

Certainly! code snippet with the implementation of the **remove_outliers_zscore** function:

```
from scipy import stats
attributes_with_outliers = ['SkinThickness', 'BMI', 'Age', 'DiabetesPedigreeFunction',
'Insulin', 'BloodPressure', 'Glucose']

def remove_outliers_zscore(df, attributes):
    for attr in attributes:
        z_scores = stats.zscore(df[attr])
        df = df[(z_scores < 3) & (z_scores > -3)]
    return df

data = remove_outliers_zscore(df, attributes_with_outliers)
```

Fig-10 Outliers Removal

The above Fig-2 which describes about the outliers removal for diabetes dataset by calculating the z-score for each feature (e.g., blood glucose level, BMI, age, etc.). Then,we would identify data points with z-scores exceeding a chosen threshold (e.g., 2 or 3). These data points would be considered outliers and could potentially be removed from the dataset to ensure that they don't unduly influence any subsequent analysis or modelling.

V.III. CORREALTION COEFFICIENT

Correlation coefficients, denoted by the lowercase letter 'r', are statistical measures used to assess the strength of the linear relationship between two variables. They quantify the degree to which changes in one variable are associated with changes in another variable.

In statistical analysis, correlations are calculated to evaluate the relationship between each column in a dataset. This process involves computing the correlation value between pairs of columns to determine how closely they are related. If two columns exhibit identical correlation values, one of them may be removed to avoid redundancy.

The correlation coefficient 'r' is computed using the formula:

$$r = \frac{m\sum ab - \sum a \sum b}{\sqrt{m\sum a^2 - (\sum a)^2} \times \sqrt{m\sum b^2 - (\sum b)^2}}$$

Where:

- 'm' indicates the quantity of data points.
- $\sum a$ indicates the sum of the values of the first variable.
- $\sum b$ represents the sum of the values of the second variable.
- $\sum ab$ denotes the sum of the product of the first and second variable values.
- $(\sum a)2$ represents the sum of the squares of the values of the first variable.
- $(\sum b)2$ indicates the sum of the squares of the values of the second variable.

The following Fig-11 table that displays a dataset's correlation coefficients between several variables

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.163658	0.217502	-0.075431	-0.068866	0.057832	0.000518	0.564932	0.248099
Glucose	0.163658	1.000000	0.231969	0.018488	0.273234	0.196347	0.091671	0.265158	0.484837
BloodPressure	0.217502	0.231969	1.000000	0.009006	-0.066151	0.272438	0.022083	0.350878	0.183155
SkinThickness	-0.075431	0.018488	0.009006	1.000000	0.460611	0.388714	0.170656	-0.140115	0.059597
Insulin	-0.068866	0.273234	-0.066151	0.460611	1.000000	0.168095	0.195227	-0.074497	0.110423
BMI	0.057832	0.196347	0.272438	0.388714	0.168095	1.000000	0.125762	0.064897	0.302966
DiabetesPedigreeFunction	0.000518	0.091671	0.022083	0.170656	0.195227	0.125762	1.000000	0.045669	0.221823
Age	0.564932	0.265158	0.350878	-0.140115	-0.074497	0.064897	0.045669	1.000000	0.256041
Outcome	0.248099	0.484837	0.183155	0.059597	0.110423	0.302966	0.221823	0.256041	1.000000

Fig-11 correlation matrix values

Figures 12, 13, and 14 delineate the correlation coefficients among variables contained within the datasets pertaining to diabetes, heart disease, and breast cancer. A positive correlation value signifies a propensity for variables to exhibit synchronous movements; an increase in one variable corresponds to a concurrent increase in the other. Conversely, a negative correlation value signifies an antagonistic relationship; an increase in one variable coincides with a decrease in the other. Values converging towards zero imply a feeble or inconsequential linear association between the variables.

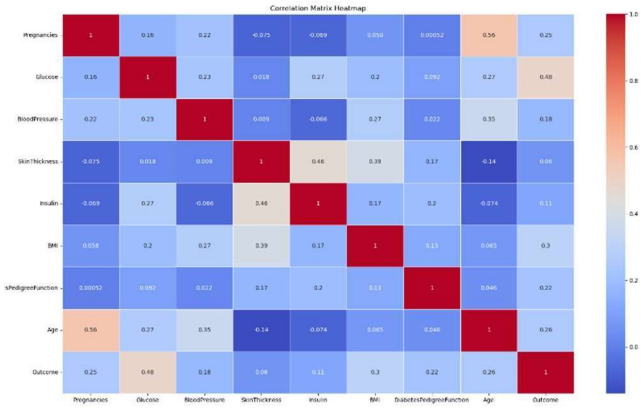

Fig-12 correlation matrix for diabetes

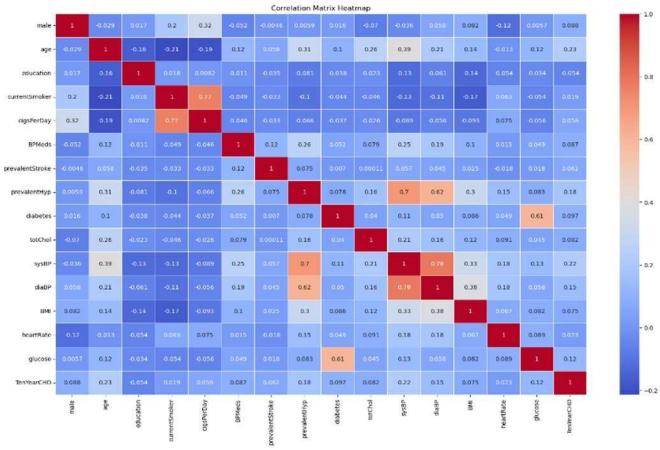

Fig-13 correlation matrix for heart

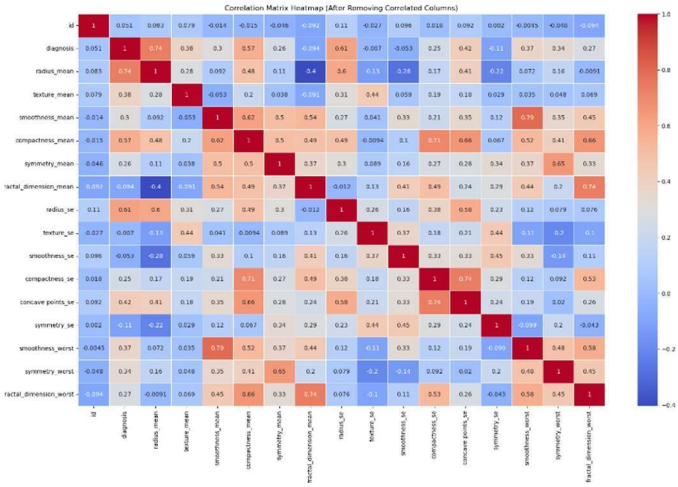

Fig-14 correlation matrix for cancer

Figure 14 displays the correlation values among the columns in the breast cancer dataset. It indicates a significant number of highly correlated columns, with correlation coefficients exceeding 0.86. To address this, we've removed the columns exhibiting such high correlations. The remaining uncorrelated columns are then presented for further analysis.

Analyzing the correlation matrix[19] helps spot variables strongly linked to each other, aiding in picking the right features for models and spotting multicollinearity issues. It's like shining a light on which variables go hand in hand. This knowledge is crucial for choosing the best predictors and avoiding problems where variables are too similar, which can mess up interpreting the model's results. By using the correlation matrix, analysts can smartly choose which variables to use, making their models more accurate and dependable.

V.IV SMOTE

In order to overcome class imbalance, machine learning uses an oversampling technique called SMOTE, which stands for Synthetic Minority Over-sampling Technique [20]. When there are substantially fewer instances of one class in the dataset than the other, it is referred to as class imbalance. When positive cases—that is, instances of a disease—are far less common than negative ones, this could occur in medical databases.In classification problems, class imbalance is a common problem where one class considerably outnumbers the other(s). Biased models that underperform on the minority class and favor the majority class may result from it.

SMOTE, a commonly employed technique, generates synthetic samples for the minority class to tackle class imbalance effectively. By doing so, it augments the model's ability to glean insights through the minority class and rectifies the skewed distribution of classes. Through the utilization of SMOTE, the dataset achieves balance in terms of class distribution, ensuring an equal number of instances for each class. Consequently, this ensures the utilization of a more representative dataset for model training, thereby enhancing performance, especially for the minority class.

SMOTE should be used carefully though, as creating synthetic samples could tamper with or add bias to the dataset. It is essential to evaluate the model's performance on simulated data and consider alternative strategies, such as adjusting class weights or employing resampling techniques, to tackle class imbalance.

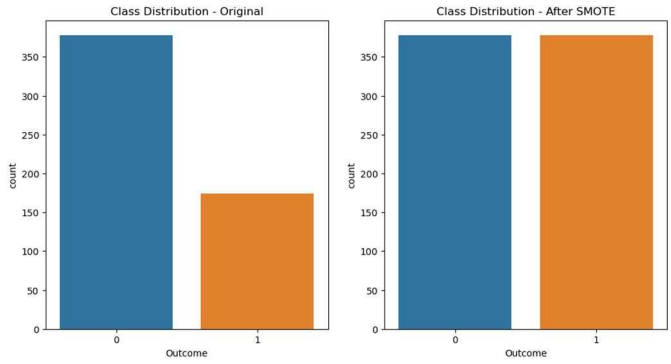

Fig-15 -class balancing using smote for diabetes

The class distribution is probably depicted in Figure 15 both before and after the (SMOTE) was used to resolve the imbalance in classes. Initial Distribution of Classes: The diabetes dataset is divided into two classes, class 0 and class 1, prior to applying SMOTE. There are 275 examples in class 0 and 125 in class 1. Class 0 is the majority class and class 1 is the minority class, indicating an imbalance in the classes.Class Distribution Following SMOTE: The class distribution has been balanced following the application of SMOTE. At this point, class 0 and class 1 each have 275 instances. In order to improve the minority class's representation in the dataset, SMOTE creates synthetic samples for it—in this case, class 1. By balancing the class distribution in the dataset, SMOTE effectively addresses class imbalance and can result in more reliable and accurate classification models (see Figure 15).

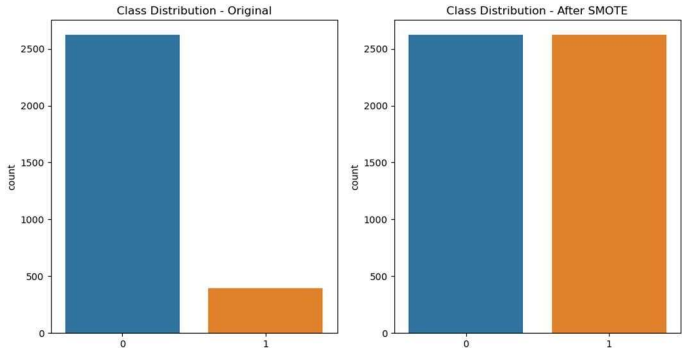

Fig-16-class balancing using smote for heart

The class distribution is shown in Figure 8 both before and after the (SMOTE) was used to resolve the imbalance in classes.Initial Distribution of Classes: Class imbalance was evident prior to SMOTE, with 475 instances in class 1 and 2750 instances in class 0. Class Distribution Following SMOTE: Class 0 and Class 1 each have 2750 occurrences following SMOTE, resulting in a balanced class distribution. In order to accomplish this, SMOTE creates artificial samples for the minority class (class 1) that are the same size as the majority class (class 0).

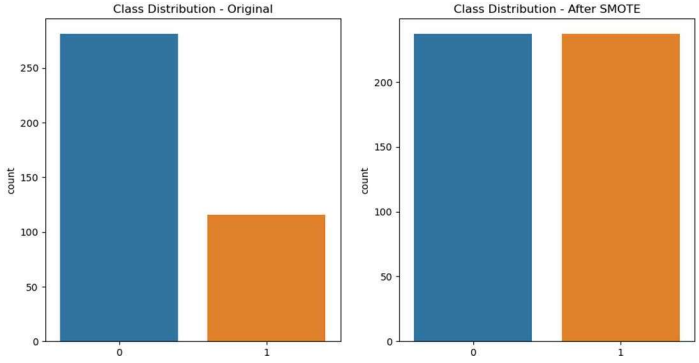

Fig-17- class balancing using smote for cancer

The class distribution is shown graphically in Figure 17 both before and after the (SMOTE) was used to address the imbalance in classes. Initial Distribution of Classes: Class 0 is the dominant class, and class 1 is the minority class. Prior to applying SMOTE, there were 275 instances in class 0 and 125 instances in class 1, suggesting a severe class imbalance.Class Assignment Following SMOTE: Class 0 and class 1 each have 275 occurrences after using SMOTE, thereby attaining a balanced class distribution. To do this, SMOTE creates artificial samples for the minority class (1) until its size is equal to that of the majority class (0).

V.V Confusion matrix

The confusion matrix is a valuable tool for assessing classification problems. It provides a comprehensive overview of the actual and predicted classifications made by a classification algorithm. The matrix

consists of rows representing the actual classes and columns representing the predicted classes, with each cell containing the count of instances where the actual class corresponds to the row and the predicted class corresponds to the column.

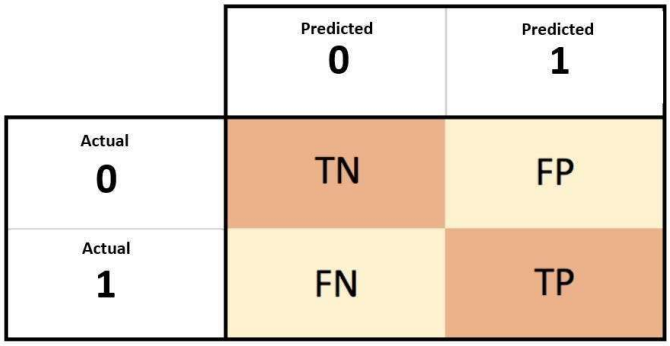

Fig-18 Confusion matrix

Figure 18 illustrates the definitions of these metrics:
True Positive (TP): Instances where the classifier correctly identifies the positive class.
True Negative (TN): Instances where the classifier correctly identifies the negative class.
False Positive (FP): Instances where the classifier incorrectly predicts the positive class.
False Negative (FN): Instances where the classifier incorrectly predicts the negative class.

Confusion matrices produced by the Random Forest algorithm applied to datasets pertaining to diabetes, heart disease, and cancer, respectively, are most likely shown in Figures 11, 12, and 13.Confusion matrices offer valuable information about a classification model's performance, especially with regard to its accuracy in classifying instances belonging to distinct classes. They are particularly helpful in comprehending the different kinds of mistakes the model makes. Making educated choices regarding model enhancements or modifications, such as adjusting thresholds, resolving class imbalances, or altering the model architecture, is possible with the use of the confusion matrix analysis.

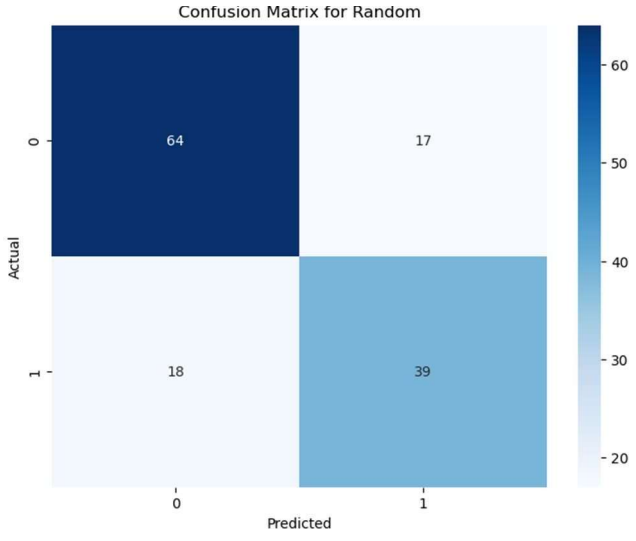

Fig-19 Confusion Matrix for Diabetes

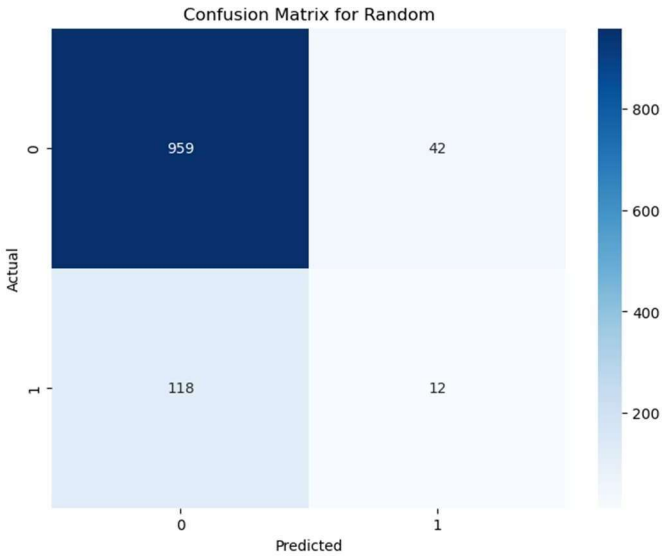

Fig-20 Confusion Matrix for Heart

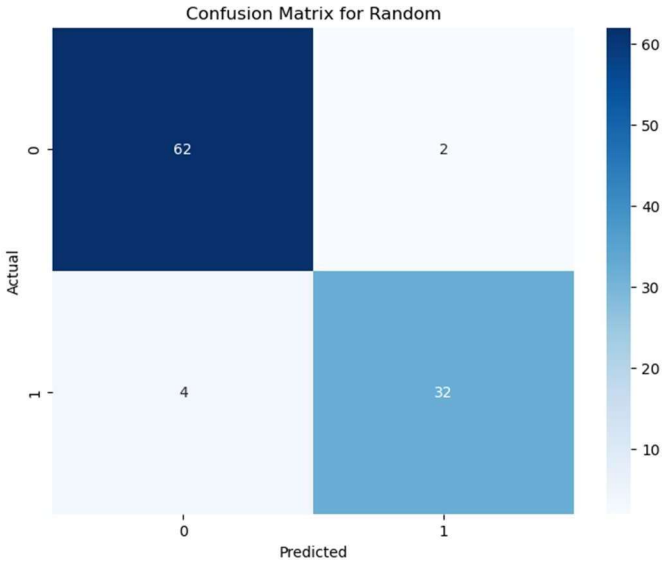

Fig-21 Confusion Matrix for cancer

These confusion matrices would show, in the context of Figures 11, 12, and 13, how well the Random Forest algorithm is working on the datasets for diabetes, heart disease, and cancer, respectively. They would facilitate the evaluation and improvement of the classification models by enabling the assessment of model accuracy, precision, recall, and other performance measures.

CLASSIFICATION REPORTS

A classification report serves as a fundamental tool in machine learning to assess the effectiveness of a classification model. It furnishes a detailed overview of various performance metrics for each class within a classification task.

A. Classification report For Diabetes

Classification Report for Naïve Bayes Model:

	precision	recall	f1-score	support
0	0.80	0.83	0.81	138
1	0.62	0.58	0.60	69
accuracy			0.74	207
macro avg	0.71	0.70	0.71	207
weighted avg	0.74	0.74	0.74	207

Fig-22 Metrics using naive bayes

The report presents various metrics to assess the model's performance:
Precision: It measures the accuracy of positive predictions. For example, precision for class "0" is 0.80, indicating that 80% of predictions for not buying were correct.
Recall: It gauges the model's ability to correctly identify actual positive cases. For class "0", recall is 0.83, meaning the model accurately predicted 83% of those who did not buy.
F1-Score: A harmonic mean of precision and recall, offering a balanced assessment.
Support: The count of data points in each class.
Overall metrics include:
Accuracy: Proportion of correct predictions; here, the model's accuracy is 0.71, indicating 71% correct classifications.
Macro Avg: Average of precision and recall across all classes.
Weighted Avg: Precision and recall averages weighted by the number of data points in each class.
Higher values are generally preferred for these metrics, but there's no universal threshold. Evaluation should consider all metrics alongside the specific task.

Classification Report for Random Forest Model using smote:

	precision	recall	f1-score	support
0	0.78	0.79	0.79	81
1	0.70	0.68	0.69	57
accuracy			0.75	138
macro avg	0.74	0.74	0.74	138
weighted avg	0.75	0.75	0.75	138

Fig-23 Metrics using random forest with smote

The classification report Fig-23 for a random forest model using SMOTE to address class imbalance includes essential performance metrics:
Precision: For instance, precision for class "0" is 0.78, indicating 78% of predictions for class "0" were correct.
Recall: For class "0", recall is 0.79, suggesting the model correctly predicted 79% of actual class "0" instances.
F1-Score: It's a harmonic mean of precision and recall, providing a balanced evaluation.
Support: Indicates the number of data points in each class; here, 81 data points in class "0" and 57 in class "1".
Overall metrics include:
Accuracy: Proportion of correct predictions; in this case, the model's accuracy is 0.74, indicating 74% correct classifications.
Macro Avg: Average of precision and recall scores for each class.
Weighted Avg: Precision and recall averages weighted by the number of data points in each class.
While higher values are generally preferred, there's no universal threshold for these metrics. Evaluating a classification model requires considering all metrics together, alongside the specific task. Notably, SMOTE can enhance model performance by addressing class imbalance, but it may also lead to overfitting, hindering performance on unseen data.

B. Classification report for Heart

Classification Report for Naïve Bayes Model:

	precision	recall	f1-score	support
0	0.90	0.91	0.90	1001
1	0.22	0.19	0.20	130
accuracy			0.83	1131
macro avg	0.56	0.55	0.55	1131
weighted avg	0.82	0.83	0.82	1131

Fig-24 Metrics using Naïve Bayes

The classification report Fig-24 for a Naive Bayes model is as follows:
Precision: For class "0", precision is 0.90, indicating that 90% of predictions for class "0" were correct.
Recall: For class "0", recall is 0.91, meaning the model correctly identified 91% of actual class "0" instances.
F1-Score: It's a harmonic mean of precision and recall, providing a balanced assessment.
Support: Indicates the number of data points in each class; 1001 data points in class "0" and 130 in class "1".
Overall metrics:
Accuracy: The model's accuracy is 0.56, indicating 56% correct classifications. However, this should be considered in the context of class balance.
Macro Avg: Average of precision and recall scores for each class.
Weighted Avg: Precision and recall averages weighted by the number of data points in each class.

Classification Report for Random Forest Model:

	precision	recall	f1-score	support
0	0.89	1.00	0.94	1001
1	0.50	0.03	0.06	130
accuracy			0.89	1131
macro avg	0.69	0.51	0.50	1131
weighted avg	0.84	0.89	0.84	1131

Fig-25 Metrics using random forest

The classification report Fig-25 for a Random Forest model is as follows:
Precision: For example, the precision for class "1" is 0.50, indicating that 50% of predictions for class "1" were correct.
Recall: For class "1", recall is 0.03, meaning the model correctly identified only 3% of actual class "1" instances.
F1-Score: It's a harmonic mean of precision and recall, reflecting the balance between them. A low F1 score (0.06) indicates poor balance for class "1".
Support: This indicates the number of data points in each class. However, the specific counts for each class are not provided.
Overall metrics:
Accuracy: The model's accuracy is 0.69, meaning it correctly classified 69% of the data points. However, the class imbalance should be considered, as it might excel at predicting the majority class but struggle with the minority class.
Macro Avg: This is the average of precision and recall scores for each class, which can be misleading with class imbalance.
Weighted Avg: This is a weighted average of precision and recall scores for each class, providing a more accurate measure of performance when there's class imbalance.
While higher values are generally preferred, there's no universal threshold for these metrics. Evaluating a classification model involves considering all metrics together and understanding the specific task and context.

C. Classification report for Breast cancer

Classification Report for Naïve Bayes Model:

	precision	recall	f1-score	support
0	0.87	0.94	0.90	64
1	0.87	0.75	0.81	36
accuracy			0.87	100
macro avg	0.87	0.84	0.85	100
weighted avg	0.87	0.87	0.87	100

Fig-26 Metrics Using Naïve Bayes

Here's a classification report for a Naive Bayes model:
Precision: For example, the precision for class "0" is 0.87, indicating that 87% of the predictions for class "0" were correct.
Recall: For class "0", recall is 0.94, meaning the model correctly identified 94% of the actual class "0" instances.
F1-Score: This is a harmonic mean of precision and recall, providing a balanced measure of both metrics. The F1-score for class "0" is not provided in the information.
Support: This indicates the number of data points in each class. In this case, there are 64 data points in class "0" and 36 data points in class "1".
Overall metrics:
Accuracy: The model's accuracy is 0.87, indicating that it correctly classified 87% of the data points.
Macro Avg: This is the average of the precision and recall scores for each class. It provides an overall view of the model's performance across classes.
Weighted Avg: This is a weighted average of the precision and recall scores for each class, where the weights are the number of data points in each class. It provides a more accurate measure of overall performance when there's class imbalance.
While higher values are generally preferred for each metric, the best evaluation involves considering all metrics together and understanding the specific task and context. For example, in scenarios where the cost of misclassification differs between classes, the focus may shift to optimizing specific metrics like recall for certain classes, even if it leads to a decrease in overall accuracy.

Classification Report for Random Forest Model using smote:

	precision	recall	f1-score	support
0	0.94	0.97	0.95	64
1	0.94	0.89	0.91	36
accuracy			0.94	100
macro avg	0.94	0.93	0.93	100
weighted avg	0.94	0.94	0.94	100

Fig-27 Metrics using Random Forest with smote

Here's a classification report for a random forest model that used SMOTE:
Precision: For example, the precision for class "1" is 0.94, indicating that 94% of the predictions for class "1" were correct.
Recall: For class "1", recall is 0.89, meaning the model correctly identified 89% of the actual class "1" instances.
F1-Score: This is a harmonic mean of precision and recall, providing a balanced measure of both metrics. The F1-score for class "1" is 0.91, indicating a good balance between precision and recall for class 1.
Support: This indicates the number of data points in each class. In this case, there are 64 data points in class "1" and 36 data points in class "0". This highlights a class imbalance, where there are more data points in class "0" than class "1".
Overall metrics:
Accuracy: The model's accuracy is 0.94, indicating that it correctly classified 94% of the data points. However, considering the class imbalance, it's essential to interpret accuracy cautiously.
Macro Avg: This is the average of the precision and recall scores for each class. It gives equal weight to each class, regardless of the number of data points. However, it can be misleading in the presence of class imbalance.
Weighted Avg: This is a weighted average of the precision and recall scores for each class, where the weights are the number of data points in each class. It provides a more accurate measure of overall performance, especially when there's class imbalance.

When evaluating the classification report, consider the class imbalance and the potential cost of misclassification. It's essential to assess the model's performance holistically, considering precision, recall, and F1-score, along with the specific context of the classification task. If the cost of misclassifying one class is significantly higher than the other, prioritize improving the corresponding metric for that class.

To display the metrics of algorithms such as Random Forest and Naïve Bayes for diabetes, heart disease, and breast cancer datasets, including various metrics as presented in Figures 28, 29, and 30.

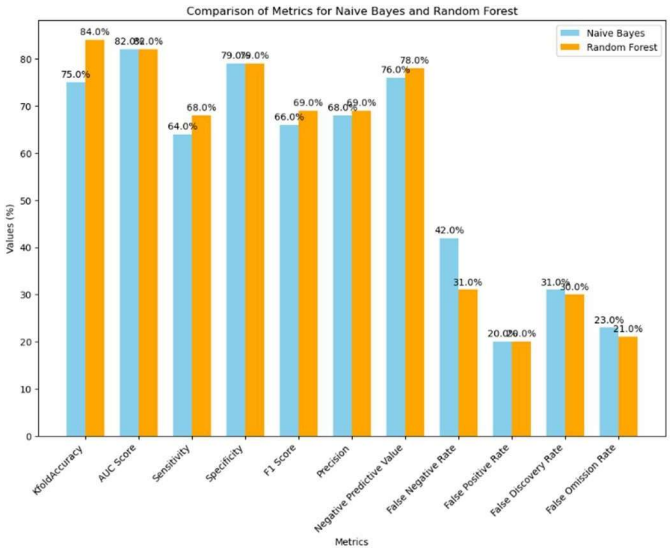

Fig-28 Metrics For Diabetes

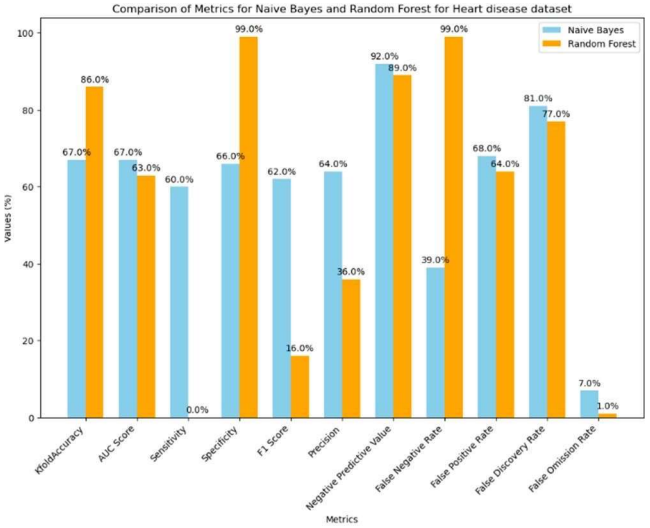

Fig-29 Metrics For Heart

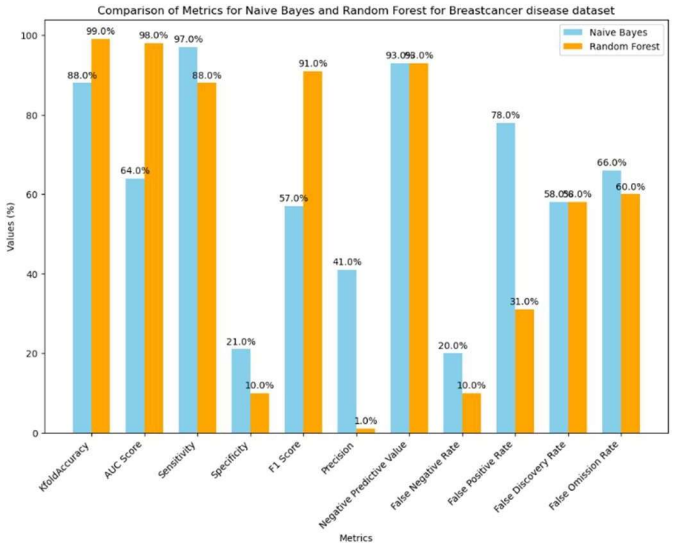

Fig-30 Metrics For Breast Cancer

VI.PERFORMANCE ANALYSIS

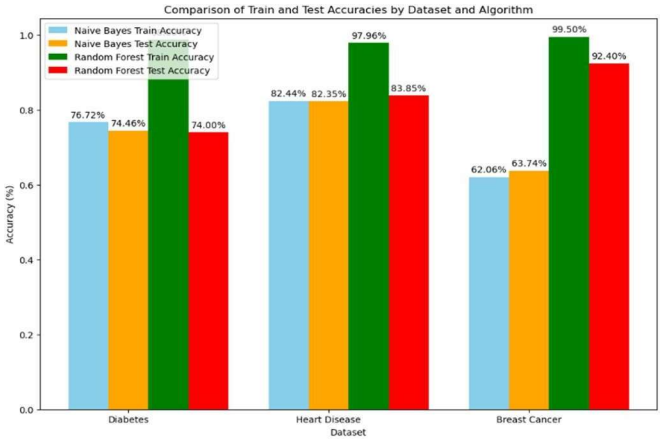

Fig-31 performance analysis of accuracies

The Naive Bayes method yields accuracies of 76.72% and 74.46% for training and test data, respectively, on the diabetes dataset, as shown visually in Figure 14. In the meantime, the Random Forest method obtains accuracy for training and test data of 98.88% and 74.03%, respectively.

Similarly, the Naive Bayes method achieves 82.44% accuracy for training data and 82.35% accuracy for test data, respectively, for the heart disease dataset. However, for training and test data, the Random Forest model achieves accuracy rates of 97.96% and 83.85%, respectively.

The Naive Bayes algorithm yields accuracies of 62.06% for training data and 63.74% for test data, respectively, for the cancer dataset. For the Random Forest method training and test data obtains accuracy of 99.50% and 92.40%, respectively.

The ROC curve shows the True Positive Rate (TPR) plotted against the False Positive Rate (FPR) across different threshold values. The classifier's overall performance is summarized by the AUC score, where a higher value suggests better performance in distinguishing between the two classes. This reflects the classifier's accuracy in correctly classifying instances across a range of thresholds.

In this scenario, we have ROC curves for Random Forest classifiers applied to three different diseases: diabetes, heart disease, and cancer. Each disease has its own ROC curve, and the corresponding AUC scores are provided:

Diabetes (AUC = 0.82): The Random Forest classifier's ROC curve (Fig. 15) for diabetes patients illustrates how well the classifier differentiates between patients with and without diabetes. The classifier performs reasonably well in this task, striking a good balance between the true positive rate and false positive rate, as evidenced by its AUC score of 0.82.

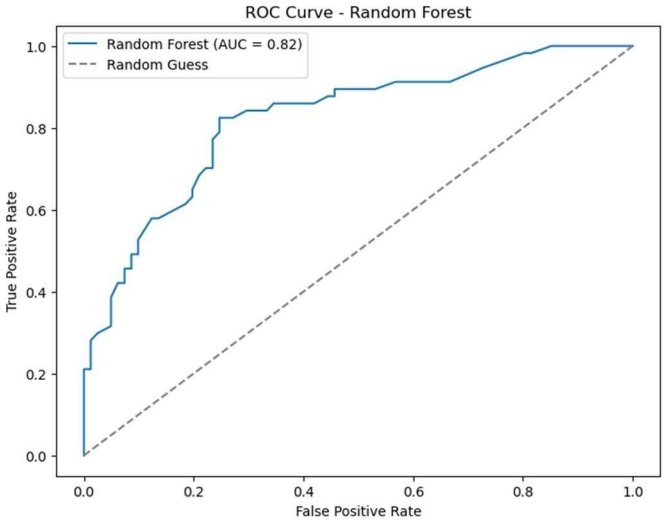

Fig-32 Roc curve of Diabetes

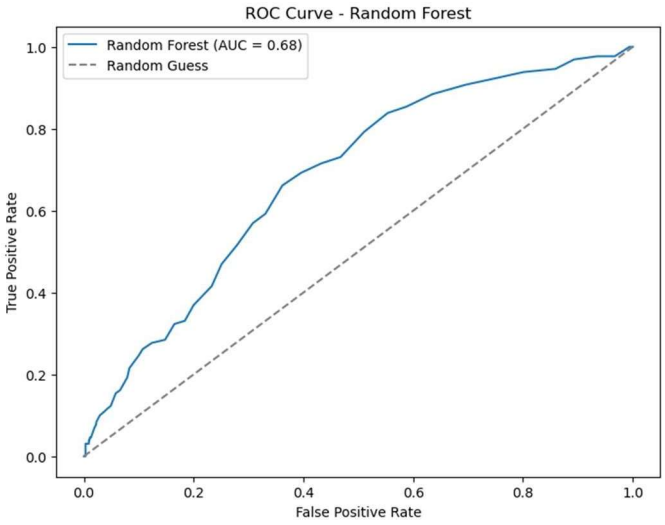

Fig-33 Roc curve of Heart

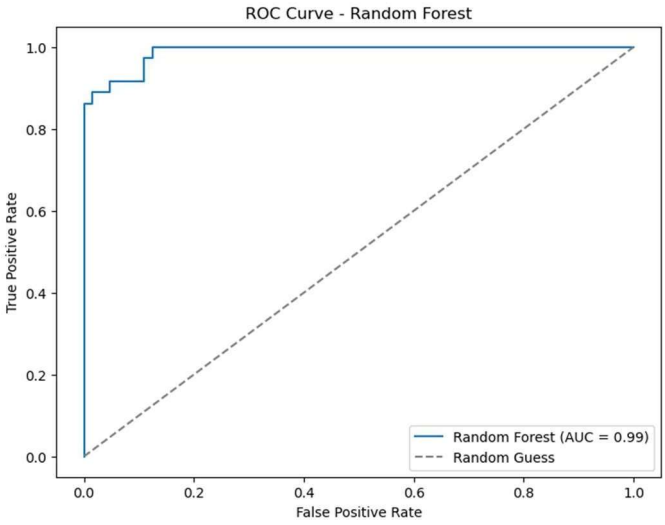

Fig-34 Roc curve of Cancer

Heart Disease (AUC = 0.62): The Random Forest classifier's performance in differentiating between people with and without heart disease is shown by the ROC curve (Fig. 33) for heart disease. The classifier performs less satisfactorily than the diabetes instance, with an AUC value of 0.62, suggesting that it has greater difficulty with this specific condition.

Cancer (AUC = 0.99): The ROC curve (Fig-34) for the Random Forest classifier applied to cancer demonstrates the classifier's ability to differentiate between cancerous and non-cancerous cases. An AUC score of 0.99 suggests exceptional performance, indicating that the classifier is highly effective in correctly classifying cancer cases.

In summary, the ROC curves provide a visual representation of the performance of the Random Forest classifier for each disease, and the AUC scores quantify the overall effectiveness of the classifier in distinguishing between positive and negative cases for each disease.

In comparison to models trained using the Naïve Bayes method, the Random Forest approach consistently produces more accurate findings when test data of every disease is applied to its trained model using classifiers. The Random Forest classifier high-dimensional property, which allows it to handle complicated datasets and catch intricate patterns in the data, is responsible for its higher performance on the test data.

The Fig-35 reported accuracies for different diseases using Naive Bayes (NB) and Random Forest (RF) classifiers as Follows:

Diabetes:
Naive Bayes: The accuracy slightly increased from 74% to 75%, while the accuracy of Random Forest significantly increased from 74% to 84%. This suggests that the Random Forest classifier performed better than Naive Bayes in classifying diabetes cases in our experiment.

Heart Disease:
Naive Bayes: The accuracy decreased from 82% to 69%, whereas the accuracy of Random Forest increased from 83% to 86%. In this case, both classifiers showed improvements, but Random Forest outperformed Naive Bayes with a higher accuracy.

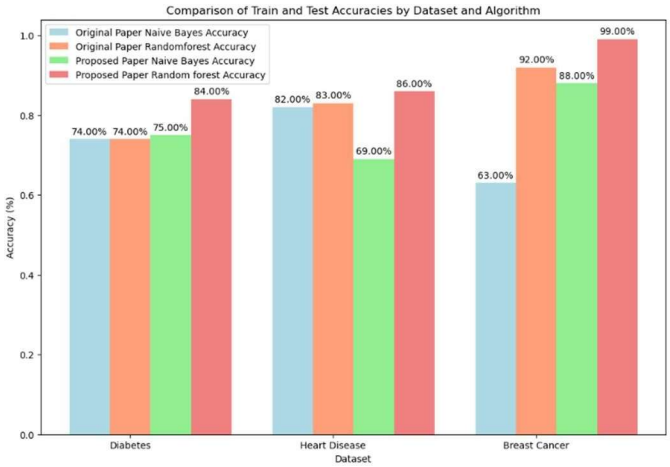

Fig-35 Comparison of Accuracies with Existing Paper

Cancer:
Naive Bayes: The accuracy notably increased from 63% to 88%, while the accuracy of Random Forest also increased significantly from 92% to 99%. Both classifiers showed substantial improvements, with Random Forest achieving the highest accuracy in our experiment.

In summary, our experiment generally resulted in higher accuracies compared to the reported accuracies in the base paper. Random Forest consistently outperformed Naive Bayes in all three diseases based on the accuracies obtained in our experiment. However, it's essential to consider other factors such as dataset characteristics, preprocessing methods, and model parameters when interpreting these results.

USER INTERFACE

To create a user interface for predicting diabetes, heart disease, and breast cancer, you can develop a web-based or mobile application that allows users to input relevant details such as medical history, symptoms, age, gender, lifestyle factors, and possibly undergo certain tests like blood pressure, blood sugar levels, cholesterol levels, etc. Based on this input, the application will then utilize machine learning models trained on historical data to predict the likelihood of the user having diabetes, heart disease,orbreastcancer

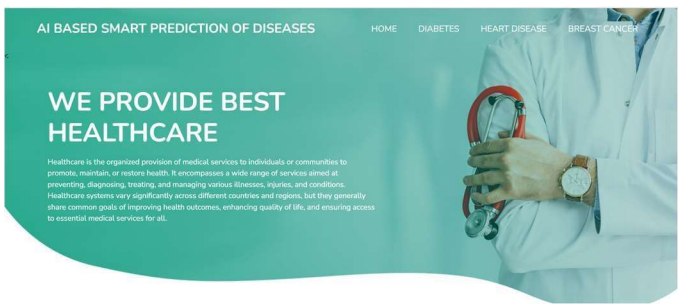

Fig-36 Home Page

Fig-37 Predicting Diseases

Fig-38 About us

Diabetes.html

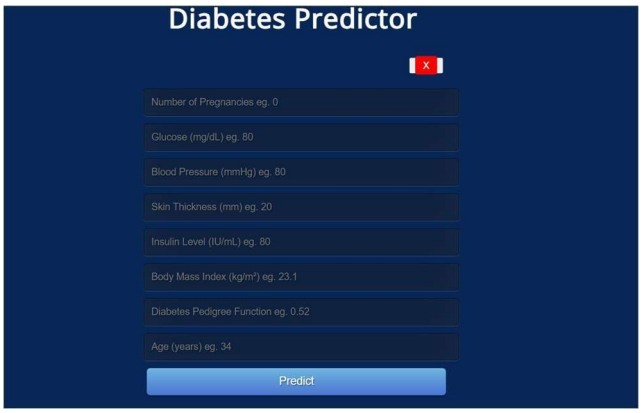

Fig-39 diabetes prediction form

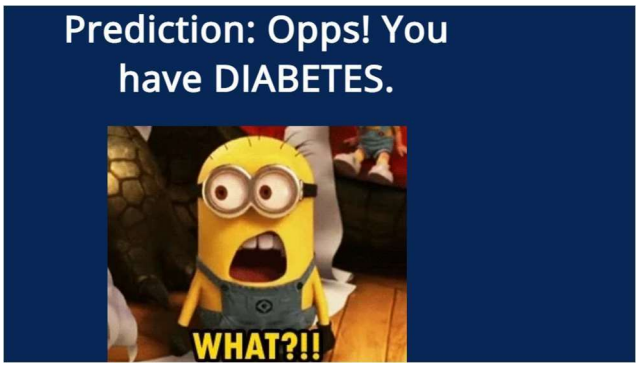

Fig-40 Showing Result having diabetes

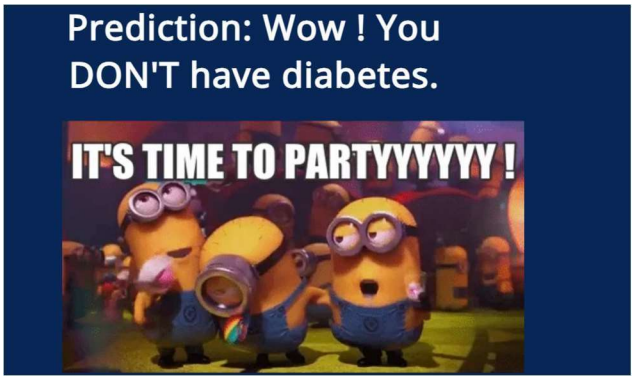

Fig-42 Showing Results that don't have diabetes

Heart.html

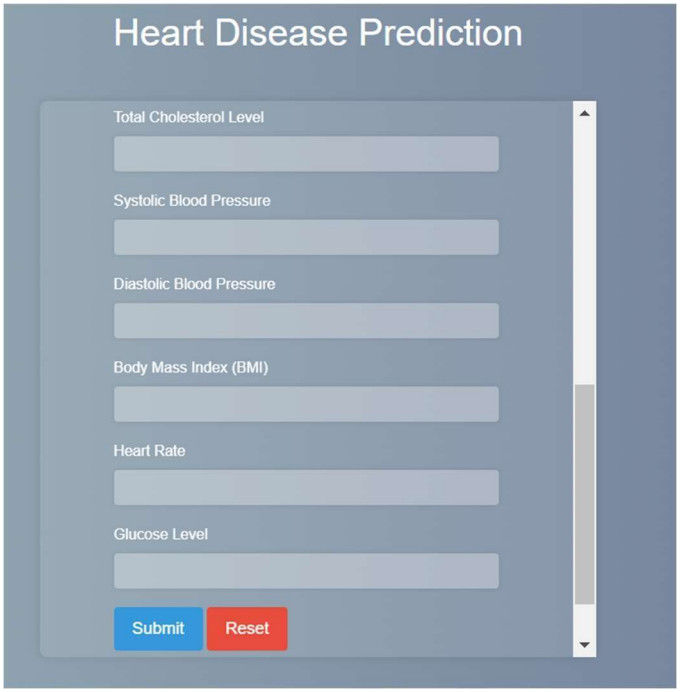

Fig-43 Heart Disease Prediction Form

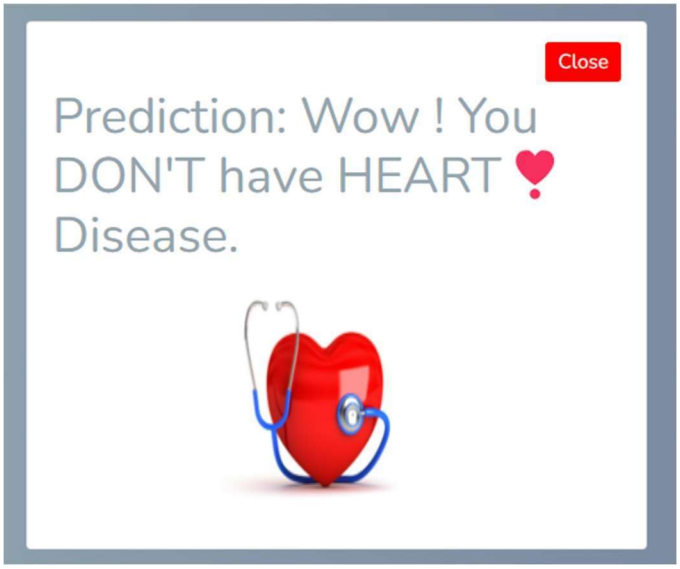

Fig-44 showing result that don't having Heart disease

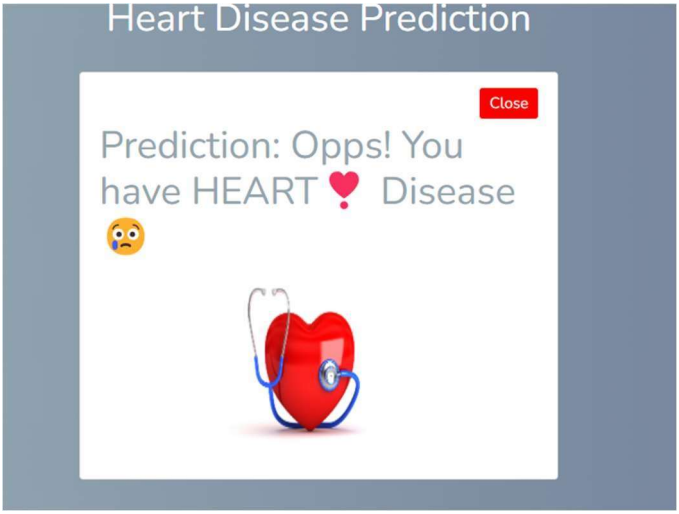

Fig-45 Showing results that having heart disease

BreastCancer.html

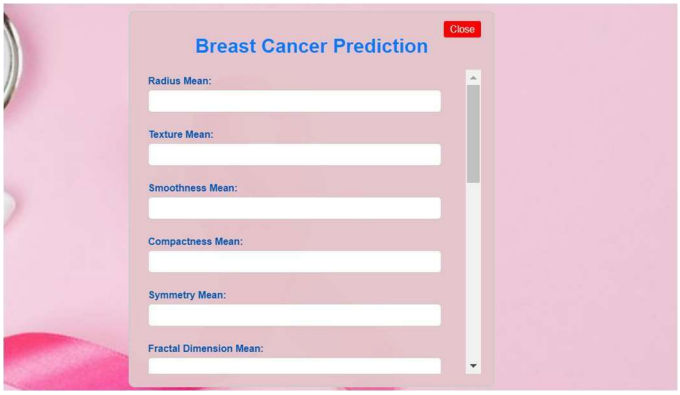

Fig-46 Breast Cancer Prediction Form

Fig-47 showing results that don't have cancer

VII CONCLUSION

In conclusion, our study demonstrates the effectiveness of data mining techniques, particularly Bayesian and Random Forest classifiers, in diagnosing diseases such as diabetes,heart disease and breast cancer. After preprocessing the datasets to ensure data quality, we trained the classifiers and evaluated their performance.

The Bayesian Classification network achieved mean accuracies of 75%, 69%, and 88% for diabetes, heart disease, and breast cancer data, respectively. In contrast, the Random Forest model exhibited significantly higher accuracy rates of 84%, 97%, and 99% for the same diseases. These results were particularly notable when the Random Forest model was combined with SMOTE.

Therefore, in the context of medical diagnosis, employing Random Forest classifiers with appropriate preprocessing techniques can lead to more reliable and accurate diagnostic models. With accuracies ranging from 84% to 99%, these models hold promise for enhancing disease detection and improving patient outcomes in clinical settings.

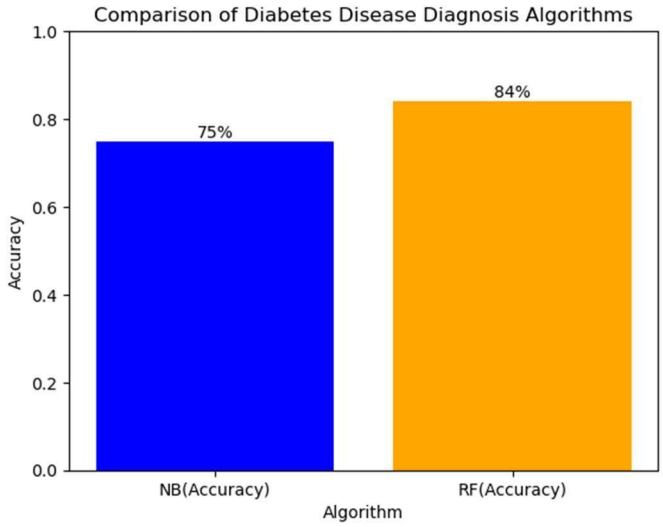

Fig-48 Algorithm Performance Comparison for Diabetes

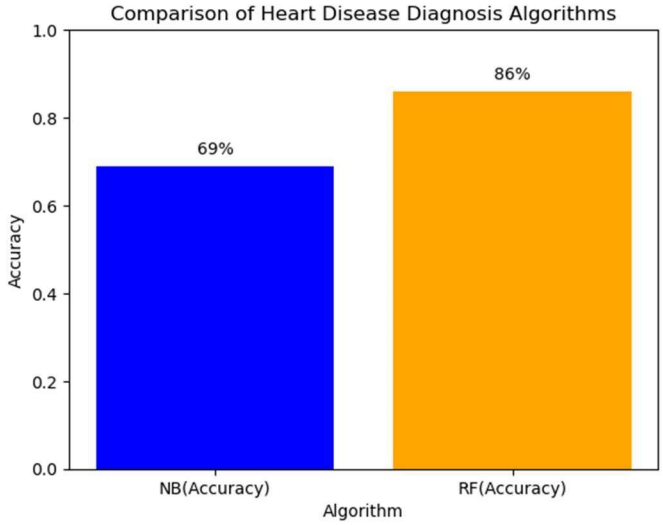

Fig-49 Algorithm Performance Comparison for Heart

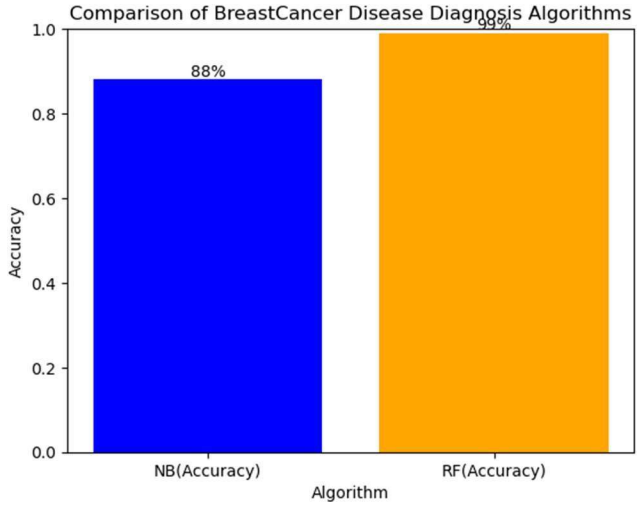

Fig-50 Algorithm Performance Comparison for Cancer

In summary, as compared to the Naïve Bayes classifier, the suggested method using Random Forest classification yields noticeably improved disease identification results (Figs. 48,49, 50). Insights into the usefulness of these models in clinical contexts can be gained by additional analysis and validation, which will ultimately enhance patient outcomes.

In conclusion, the Random Forest algorithm is the recommended option for disease classification jobs because it performs the best in accuracy on both training and test data across all three disease datasets.

When comparing the accuracy results, it is clear that for all three diseases, the Random Forest model performs better than the Naïve Bayes classifier. Based on the provided datasets, this suggests that Random Forest is more successful in correctly identifying diabetes, coronary heart disease, and cancer.

VIII FUTURE WORKS

HyperparameterTuning:

Methodically tuning the hyperparameters of selected AI algorithms is essential to optimize accuracy and performance. This process involves exploring various hyperparameter configurations for each algorithm, typically through techniques such as grid search, random search, and Bayesian optimization. By systematically adjusting these parameters, the algorithm can be fine-tuned to achieve the best possible performance on a given task or dataset

Clinical Trials and Validation Studies:

Thorough clinical trials and validation studies play a vital role in evaluating the therapeutic value and impact of developed models. These studies generate empirical evidence showcasing the effectiveness of the models in enhancing patient outcomes, diagnostic accuracy, and healthcare resource utilization. By conducting rigorous assessments in real-world clinical settings, the validity and reliability of the models can be established, ensuring their practical applicability and positive impact on patient care.

IX REFERENCES

1. K. Modi, I. Singh, Y. Kumar, A comprehensive analysis of artificial intelligence techniques for the prediction and prognosis of lifestyle diseases. Arch. Computat. Methods Eng. (2023)
2. Goyal, A., Gupta, Y., Katiyar, S., Misra, A., & Vikram, N. K. (2021). Lifestyle modification in the management of obesity, diabetes, hypertension, dyslipidemia, and cardiovascular disease: a review. Indian Journal of Endocrinology and Metabolism, 25(6), 441.
3. Dong, X., & Yang, J. (2022). Research on the Application of Data Mining Technology in the Field of Medical Health. In 2022 7th International Conference on Control, Robotics and Cybernetics (CRC) (pp. 643-646). IEEE.
4. Mohammed, M. A., Al-Hajj, S. A., & Al-Azzawi, A. M. (2023). A Comparative Study of Machine Learning Algorithms for Disease Prediction. International Journal of Scientific & Engineering Research, 14(1), 302-308.
5. A. Jakka, J. Vakula Rani, Performance evaluation of machine learning models for diabetes prediction. Int. J. Innov. Technol. Exploring Eng. 08(11) (2019)
6. S. Karun, A. Raj, G. Attigeri, Comparative analysis of prediction algorithms for diabetes, in International Conference on Computer, Communication and Computational Sciences, IC4S, vol. 759, Kathu, Thailand (2019)
7. A. Leiherer et al., Data on the power of high betatrophin to predict cardiovascular deaths in coronary patients. Data in Brief 28(December), 104989 (2020). https://doi.org/10.1016/j.dib.2019.104989
8. C.S. Wu, M. Badshah, V. Bhagwat, Heart disease prediction using data mining techniques. ACM Int. Conf. Proc. Ser. 3(7), 7–11 (2019)
9. B. Baad, Heart disease prediction and detection. Int. J. Res. Appl. Sci. Eng. Technol. 7(4), 2293–2299 (2019)
10. C. Beyene, Survey on prediction and analysis the occurrence of heart disease using data mining techniques. Int. J. Pure Appl. Math. 118(8), 165–74 (2018)
11. J.P. Corsetti, et al., Data in support of a central role of plasminogen activator inhibitor-2 polymorphism
12. H. Dhahri, E. Al Maghayreh, A. Mahmood, W. Elkilani, M.F. Nagi, Automated breast cancer diagnosis based onmachine learning algorithms. J. Healthc. Eng. 2019, 11, ArticleID4253641. https://doi.org/10.1155/2019/4253641
13. S.A. Mohammed, S. Darrab, S.A. Noaman, G. Saake, Analysis of breast cancer detection using different machine learning techniques, in Data Mining and Big Data. DMBD 2020. Communications in Computer and Information Science eds. by Y. Tan, Y. Shi, M. Tuba, vol 1234 (Springer, Singapore, 2020). https://doi.org/10.1007/978-981-15-7205-0_10
14. M. Amrane, S. Oukid, I. Gagaoua, T. Ensar˙I,Breast cancer classification using machine learning, in 2018 electric electronics, computer science, biomedical engineerings' meeting (EBBT) (2018), pp. 1–4. https://doi.org/10.1109/EBBT.2018.8391453
15. Handling Missing Values in machine learning. https://towardsdatascience.com/working-withmissing-data-in-machine-learning-9c0a430df4ce
16. F. Shahidi, S.M. Daud, H. Abas, N.A. Ahmad, N. Maarop, Breast cancer classification using deep learning approaches and histopathology image: a comparison study. IEEE Access 8, 187531–187552(2020). https://doi.org/10.1109/ACCESS.2020.3029881

17. Cha SH, Steemers K, Kim TW (2018) Modeling space preferences for accurate occupancy prediction during the design phase. Autom Constr 93:135–147
18. Gunduz V (2020) Chapter 7—Risk management in banking sector. In: Management and Strategy. Artikel Akademi, pp 121–135
19. In-Database Machine Learning 2: Calculate a correlation Matrix—A Data Exploration Post, Vertica.https://www.vertica.com/blog/in-database-machine-learning-2-calculate-a-cor relation-matrix-a-data-exploration-post
20. M. Sireesha, S.N. Tirumala Rao, S. Vemuru, Predictive analysis of imbalanced cardiovascular disease using SMOTE. Int. J. Adv. Sci. Technol. 29(5), 6301–6311 (2020)