Prediction of Risk of Heart Attack using Machine Learning Techniques

Kartik Deogire
Department of Computer Engineering
PCET's Pimpri Chinchwad College of Engineering
Pune - 411044, India

Sahil Dhake
Department of Computer Engineering
PCET's Pimpri Chinchwad College of Engineering
Pune - 411044, India

Shreevallabh Chidrawar
Department of Computer Engineering
PCET's Pimpri Chinchwad College of Engineering
Pune - 411044, India

Dhanashree Patil
Department of Computer Engineering
PCET's Pimpri Chinchwad College of Engineering
Pune - 411044, India

ABSTRACT
The usefulness of machine learning models in forecasting the risk of a heart attack based on health-related variables is examined in this study. The classification models Gaussian Naive Bayes, K-Nearest Neighbors and Random Forest were created and assessed using performance measures like recall, accuracy, precision, F1-score. The dataset was heavily preprocessed, handling null values, duplicates, outliers, and feature transformation. It had 10 predictor variables and a target variable with 5110 observations. The most instructive elements for model training were found using feature selection approaches.

Using k-fold cross-validation for KNN and GridSearchCV for Random Forest, hyperparameter tweaking was carried out for the models on the remaining 25% of the dataset after they had been trained on 75% of it. The results show that KNN outperformed Gaussian Naive Bayes and Random Forest, with the greatest accuracy of 96.4% following hyperparameter adjustment. SMOTE was also used to improve model robustness by addressing class imbalance. In summary, this study's best model for predicting the likelihood of a heart attack was KNN. These results demonstrate how machine learning models can improve early detection and individualized patient care by advancing risk assessment and intervention tactics in the healthcare industry.

Keywords
Heart Attack Prediction, Machine Learning Models, K-Nearest Neighbors (KNN), Gaussian Naive Bayes, Random Forest, Data Preprocessing, SMOTE, Hyperparameter Tuning, Healthcare Data, Risk Prediction.

1. INTRODUCTION
The creation and assessment of Naive Bayes, K-Nearest Neighbours and Random Forest classification models for heart attack risk prediction are the main objectives of this study. These models use important health-related factors to categorize people into various risk groups. The Naive Bayes model is chosen due to its efficient management of continuous data; the predictive power of the model is statistically assessed using performance metrics including recall, accuracy, precision, F1-score.

Important health features are identified by examining feature importance and relationships. With the use of instance-based learning, the KNN model attempts to categorize people into groups according to their risk of having a heart attack. In order to accurately forecast risk, the study also uses the Random Forest classification model, which is renowned for its ensemble learning methodology. This model evaluates a wide range of health-related factors. In order to show Random Forest's potential for accurate risk assessment, the research will examine its performance utilizing classification metrics and visualizations. The study explores how certain health characteristics affect the model's ability to forecast the risk of a heart attack and how they interact with it.

The potential for this research to change risk assessment and intervention strategies in actual healthcare settings and improve public health outcomes in the fight against cardiovascular diseases is significant. Ultimately, it advances early heart attack risk assessment and personalized patient care.

2. LITERATURE REVIEW
The authors of paper[1] studied how machine learning algorithms could be used to predict heart disease. The main metric used to assess algorithm performance was accuracy, and the dataset used in the study had 14 attributes. Using a dataset from the UCI repository, the SVM showed improved prediction accuracy.
In order to meet the critical need of offering precise cardiac disease prediction, the existing method is described in the publication. Seven machine learning algorithms are used to predict heart disease: Random Forest, Decision Trees, Naïve Bayes, Support Vector Machine, Logistics Regression, Extreme Gradient Boost, and KNN. The best option is determined to be Support Vector Machine, with accuracy rates of 93.67%, 94.64%, 85.37%, 98.05%, 86.34%, 94.64%, and 87.81%, respectively[1].

In paper[2], the study focuses on predicting a patient's risk of developing heart failure using the 13 features that make up the Heart Failure Dataset. Based on a variety of risk factors, the disease's severity is analyzed and predicted using two classification algorithms: Decision Tree and Naïve Bayes. The study's 82% Decision Tree and 86% Gaussian Naïve Bayes accuracy rates demonstrate how well these machine learning models predict heart disease.
In paper[2] the effectiveness of machine learning in predicting heart diseases is highlighted by the common findings found in both the current system and the literature review. Numerous research works have illustrated the effectiveness of machine learning algorithms, including Decision Trees and Naïve Bayes, among others, in producing precise forecasts. The significance of accurate and timely disease prediction in enhancing healthcare outcomes is highlighted by these findings [2].

This paper[3] focuses on applying the probabilistic classifier Naive Bayes algorithm, which is based on Bayes' theorem, to the creation of a Heart Disease Prediction System (HDPS). Naive Bayes classifiers, with their built-in simplicity and efficiency, presume feature independence, making them a good fit for applications in medical science. The system helps forecast heart disease by computing posterior probabilities based on characteristics including blood pressure, cholesterol levels, type of chest pain, age, and sex, among others. The study uses Weka, a machine learning tool, for data mining jobs. It uses a 70% percentage split for categorization, and the results show promise. The Naive Bayes model exhibits a high accuracy rate of 86.4198% despite a limited margin of misclassification; on average, precision, recall, and F-measure are 71%, 74%, and 71.2%, respectively. These findings illustrate the algorithm's effectiveness in outperforming alternative approaches, especially when utilizing characteristics that are not clear markers of heart illness. This further emphasizes the HDPS's capacity to accurately and efficiently diagnose medical data and forecast heart ailments[3].

The methodology in paper[4] entails preparing datasets, dividing them into test and training sets, and standardizing them so that they can be analyzed. The training set is used to train SVM and Naive Bayes models, and the test set is used to assess them using predefined metrics. The study highlights the significance of early diagnosis and predictive modeling in reducing heart-related risks and enhancing patient outcomes using statistical analysis and comparisons with prior research. The usefulness of Support Vector Machine (SVM) and Naive Bayes algorithms in predicting the existence of heart disease and patient survival is examined in this work. The research uses confusion matrices, ROC curves, and AUC analysis to assess model accuracy using sixteen attribute datasets that have been clinically validated. Heart illness prediction with Naive Bayes yields an accuracy of 87%, whereas heart survival prediction with SVM and Naive Bayes yields 88% and 93% accuracy, respectively. These findings highlight both systems' excellent prediction capabilities, with Naive Bayes performing especially well in survivability prediction[4].

In paper[5], the authors used the random forest data mining algorithm in this study to forecast the occurrence of heart disease. With a sensitivity value of 90.6%, specificity value of 82.7%, and accuracy value of 86.9% for heart disease prediction, the experimental work's results were impressive. Furthermore, based on the Receiver Operating Characteristic (ROC) curve's Area Under the Curve (AUC), the diagnosis rate was a remarkable 93.3%. Significantly, the study implies that the developed system can be used for purposes other than predicting heart disease. To improve predictive accuracy, the authors suggest integrating various machine learning algorithms such as Naïve Bayes, fuzzy logic, K-NN, decision trees, and linear regression. In addition, the authors suggest utilizing cloud computing to effectively handle the significant amount of patient data.The paper's literature review provides information about current research and systems in the field. Previous research has used a variety of machine learning algorithms and investigated different strategies for the prediction of heart disease[5].

This paper[6] uses data from the Cleveland database to assess the predictive power of random forest and logistic regression models. From the dataset of 76 attributes, a subset of 14 attributes was chosen, and this subset was used for training and testing both models. To evaluate the efficacy of the models, performance metrics such as precision, recall, F1-score, accuracy, MSE, confusion matrix, and ROC curve were used. Preprocessing included feature significance analysis and attribute classification to identify important heart disease predictors.With precision, recall, and F1-score for the healthy class at 85.00%, 79.00%, and 82.00%, and for the sick class at 81.00%, 86.00%, and 84.00%, respectively, the logistic regression model yielded an accuracy of 83.00%. With an accuracy of 91.00%, precision, recall, and F1-score for the healthy class at 94.00%, 86.00%, and 90.00%, and for the sick class at 88.00%, 95.00%, and 91.00%, respectively, the random forest model outperformed the logistic regression model. Additionally, the random forest model performed better overall as seen by higher macro and weighted averages for precision, recall, and F1-score.In the ROC curve study, it also showed a higher AUC value, indicating improved class discrimination ability. Atypical angina and the quantity of main blood arteries were found to be significant indicators of heart disease using feature significance analysis. Overall, the findings highlight the random forest model's better predictive ability than logistic regression and highlight the significance of feature analysis and model selection in raising prediction accuracy and locating important heart disease predictors. They also point to future directions for research to improve prediction models[6].

The paper[7] investigates the predictive power of many machine learning techniques, such as Naive Bayes, Random Forest, and Decision Tree, in relation to heart disease risks. Based on probabilistic principles, Naive Bayes calculates posterior probabilities for data classification by assuming conditional independence among characteristics. By randomly selecting features and sampling subsets, Random Forest enhances performance and robustness by using ensemble techniques to generate several decision trees. Using entropy and information gain for informative feature selection and decision making, Decision Tree builds a tree-like structure to represent data. While Random Forest initially achieved an accuracy of 81.53%, feature selection led to a significant improvement to 88.18% and 90.52% for distinct feature sets. These approaches have demonstrated promising accuracy. Similar improvements were shown by Multilayer Perceptron, whose accuracy increased with feature selection from 78.52% to 96.18%. The potential of machine learning to transform the detection and treatment of cardiovascular disease is highlighted by these findings[7].

Using the KNN approach in data mining, the paper[8] uses M2M technology and simplified criteria to predict heart illness in remote patient monitoring. With just 8 criteria, cardiac disease prediction attains 81.85% accuracy, which is equal to employing the recommended 13. While Naive Bayes achieves 74.49% with 8 parameters and 79.93% with 13, Decision Tree (Simple CART) obtains 80.27% accuracy with 8 parameters and 79.93% with 13. These results demonstrate the promise of KNN for remote patient monitoring, especially in resource-constrained environments like Indonesia, and point to the effectiveness of simpler parameters in prediction models. When looking for trustworthy predictive models to manage heart disease, healthcare professionals can benefit greatly from these insights[8].

In paper[9], the study describes a machine learning algorithm-based heart disease prediction system that makes use of logistic regression and KNN. Based on their medical history, it attempts to precisely forecast which patients are most likely to receive a heart disease diagnosis. With many classifiers, the model outperforms earlier systems with a single data mining technique, with an accuracy of 87.5%. KNN and logistic regression both perform well; KNN's accuracy of 88.52% is the greatest[9].

Paper[10]. The prevalence of heart disease can be considerably decreased by identifying its signs early. In this regard, our

suggested solution uses a variety of machine learning techniques to forecast the onset of cardiac disease beforehand. The algorithms take into account several input features such as age, sex, presence of chest pain, maximal heart rate, exercise-induced angina, old peak, slope, number of main vessels, cholesterol level, fasting blood sugar, resting ECG status, and Thalassemia indicators. For prediction, ten classifiers are used: weighted voting, LR, NB, SVM, DT classifier, AdaBoost, QDA, K neighbors, GB classifier, and XGB. Weighted voting stands out among these because of its higher AUC value, which indicates that it is a more accurate classifier than the others. As a result, the best technique for determining heart disease symptoms is weighted voting. With early understanding of the symptoms of cardiovascular illness, this technology has enormous potential for use in the medical industry, enabling medical professionals and doctors to make informed decisions[10].

3. METHODOLOGY

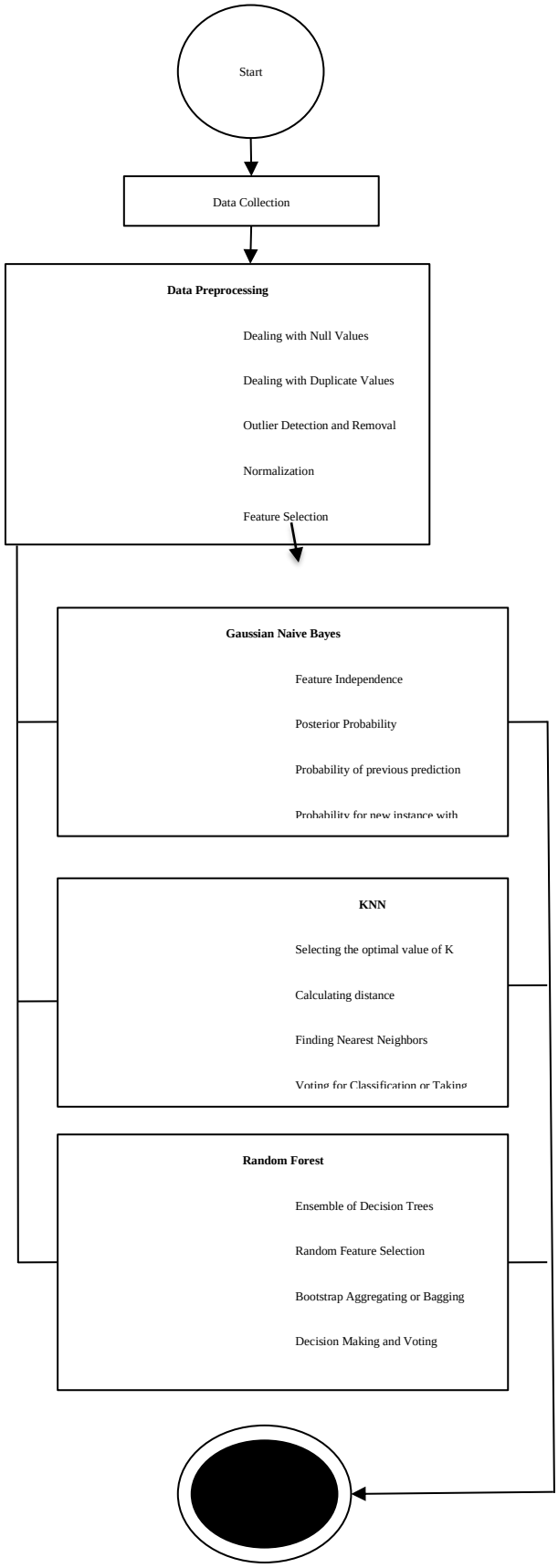


Fig. 1: Architecture Design

3.1 Dataset Analysis

'Healthcare-dataset-stroke-data.csv' dataset is used in this study. It has in total 11 columns in which 10 are a predictor variable and a target variable. It has 5 categorical variables and 6 numerical variables. There are 5110 observations.

Table 1. First Five Observation

gender	age	hypertension	heart_disease	ever_married	work_type
Male	67	0	1	Yes	Private
Female	61	0	0	Yes	Self-employed
Male	80	0	1	Yes	Private
Female	49	0	0	Yes	Private
Female	79	1	0	Yes	Self-employed

Residence_type	avg_glucose_lvel	bmi	smoking_status	stroke
Urban	228.69	36.6	formerly smoked	1
Rural	202.21	nan	never smoked	1
Rural	105.92	32.5	never smoked	1
Urban	171.23	34.4	smokes	1
Rural	174.12	24.0	never smoked	1

To obtain the data types of the features in the dataset we can use the command 'df.dtypes' which provides the insight of which feature is categorical and which is numerical for example we have 'smoking_status' as categorical and 'age' as numerical.

Table 2. Data Types of Columns

Column	Type
gender	object
age	float64
hypertension	int64
heart_disease	int64
ever_married	object
work_type	object
Residence_type	object
avg_glucose_level	float64
Bmi	float64
smoking_status	object
Storke	int64

3.2 Data Preprocessing

3.2.1 Dealing With Null Values

Null values have the potential to cause inconsistent and inaccurate results, which can undermine the validity of machine learning models and statistical analyses. It is imperative to appropriately handle null values, either by imputation or removal, to guarantee that the absence of data does not impair the overall quality of the data. They have observed the distribution of the data by plotting it (bmi x count) and have used the median method to impute the feature 'bmi' which had 201 null values.

Code: df.isna().sum()
Code: df['bmi'] = df['bmi'].fillna(df['bmi'].median())

Table 3. Count of Null Values

Column	Null Count
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
Bmi	201
smoking_status	0
Storke	0

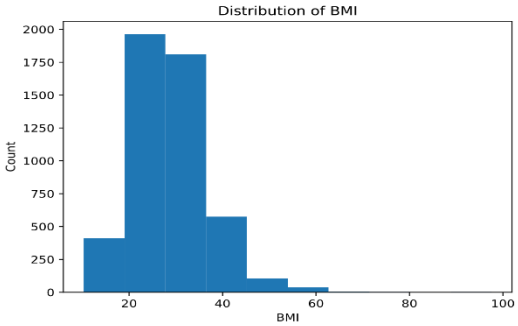


Fig. 2: Distribution of BMI

Table 4. Count of Null Values after applying Median

Column	Null Count
gender	0
age	0
hypertension	0
heart_disease	0
ever_married	0
work_type	0
Residence_type	0
avg_glucose_level	0
Bmi	0
smoking_status	0
Storke	0

3.2.2 Dealing with Duplicates

There were no duplicates present in the dataset

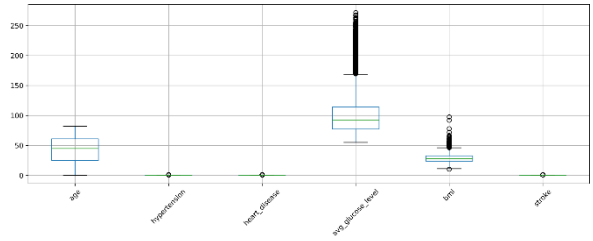
Code: df.duplicated().sum()

3.2.3 Dealing with Outliers

Identification of outliers in a dataset is important for a number of reasons. Data points known as outliers, or those that differ significantly from the majority, can have a significant impact on a number of different aspects of statistical analyses and machine learning models. Their existence may have an outsized impact on model parameters and forecasts, which could skew and produce inaccurate results from machine learning algorithms as well as statistical analysis. It is critical to identify and handle outliers in order to have more accurate, significant, and trustworthy insights. Finally, anomalies have the potential to destroy data consistency and integrity. In order to maintain data integrity and make sure that analyses and models are based on reliable and trustworthy foundations, outliers must be addressed, either by removal or appropriate handling.

In this dataset they have found outliers in ‘avg_glucose_level’ and ‘bmi’ using boxplot. They have used “Interquartile range” method to deal with this outlier where we take the first and third quartile of the dataset and the difference of the third and first quartile provides us with IQR which is then multiplied by 1.5 and then to have lower bound this value (IQR *1.5) is subtracted from the first quartile and for the upper bound they add this (IQR*1.5) to the third quartile. And then any value below lower bound and any value above the upper bound is considered outlier and removed.

Fig. 3



Boxplot of Features for Outliers Detection

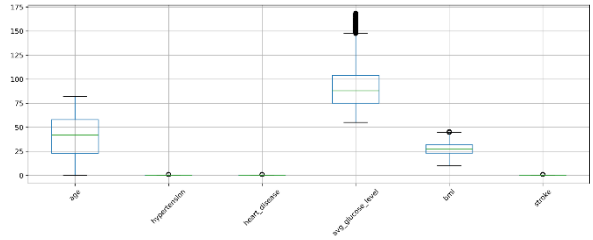


Fig. 4: Boxplot of Features after Removal of Outliers

3.2.4 Feature Transformation for normal distribution

The process of transforming a dataset's features to bring their distributions closer to a Gaussian (normal) distribution and more symmetrical is known as feature transformation. Considering that many statistical techniques assume regularly distributed data, this can be advantageous for statistical modeling and inference.

In this study features such as ‘age’ and ‘bmi’ were normally distributed; but the ‘avg_glucose_level’ was Right Skewed.

The ‘avg_glucose_level’ variable in the dataset used was preprocessed using feature transformation in this study. This is a standard machine learning technique for improving model performance, particularly with skewed data distributions. The ‘np.log1p()’ function was used to apply the natural logarithm transformation to each value in the ‘avg_glucose_level’ column, making use of the NumPy library. In order to ensure numerical stability and avoid undefinable outcomes for zero or negative values, this algorithm involves adding 1 to each value in advance. The adjustment attempts to promote symmetry and approximate a normal distribution by reducing skewness in the 'avg_glucose_level' distribution. The data is better suited for statistical analysis and modeling methods that work best with normally distributed data when they are normalized. All things considered, using np.log1p() to apply feature transformation is an essential preprocessing step in our research that improves the accuracy of our findings and machine learning model optimization.

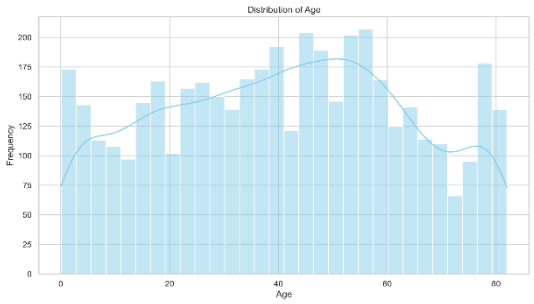


Fig. 5: Distribution of Age

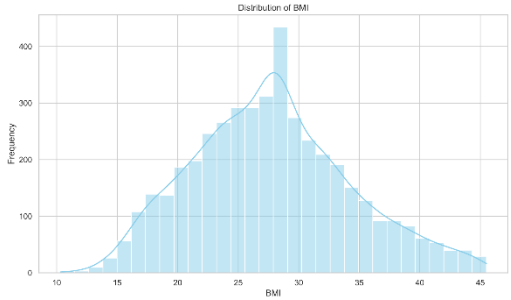


Fig. 6: Distribution of BMI

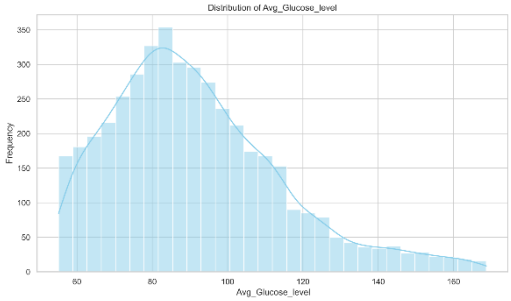


Fig. 7: Distribution of Avg_Glucode_Level

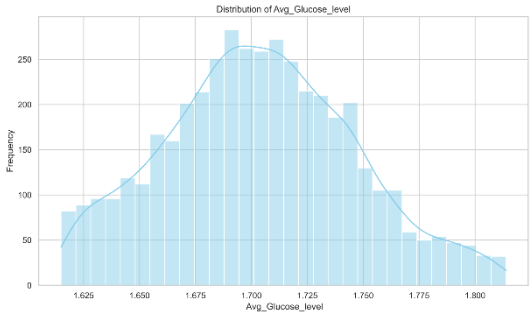


Fig. 8: Distribution of Avg_glucose_level after Transformation

3.2.5 Heatmap

An illustration of the relationships between variables, usually presented as a color-coded matrix, is called a correlation heat map. The heat map uses a color gradient to show the strength and direction of correlations between variables. Particularly in complex data sets, these visual aids are indispensable for data analysis as they facilitate the rapid identification of variable interconnections and their intensities. In this paper they have plotted the heatmap for the numerical variables ‘age’, ‘bmi’ , ‘avg_glucose_level’.

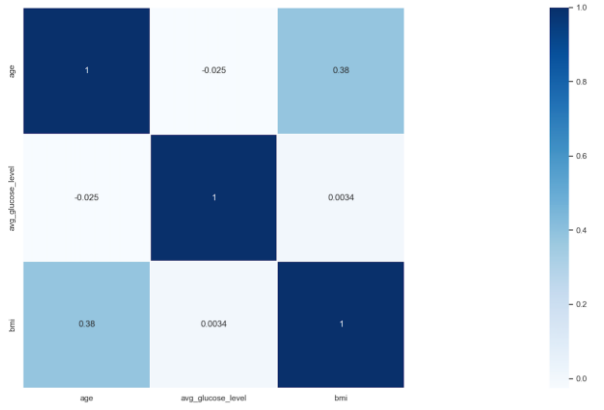


Fig. 9: Heatmap

3.2.6 Feature Selection

A crucial preprocessing stage in data analysis is feature selection, which selects a subset of pertinent features from a broader pool inside a dataset. Its main objective is to maximize model performance, minimize overfitting, and improve interpretability by eliminating redundant features and concentrating on the most informative ones. According the research down for this paper, feature selection is essential for optimizing the efficiency of predictive algorithms and for speeding the modeling process. The paper's goal is to create models with the best possible predictive accuracy and generalization abilities by carefully choosing the features that are most pertinent to the target variable. Furthermore, by removing superfluous features, we reduce the possibility of overfitting and produce predictions that are more trustworthy and stable. By making it easier to see the underlying patterns and correlations in the data, feature selection enables researchers to draw conclusions from the data that can be put into practice.

Use of Chi-squared test to assess the correlation between ‘residence_type’ and the target variable, ‘stroke’ can be seen in this research . The Chi-squared test resulted in a p-value of 0.605 at the chosen significance level ($\alpha = 0.05$), indicating that there is no statistically significant correlation between the target variable and dwelling type. This suggests that ‘resident_type’ may not be a major predictor of the target variable in the study. As a result, it has been dropped.

To expedite the modeling procedure, they have used threshold variance as a feature selection strategy in the research. Using this method, low variance features—which usually contain little information relevant for modeling tasks—are filtered out. Because of this dimensional reduction is successfully implemented , simplifying the model and potentially improving performance by minimizing noise and overfitting, by establishing a threshold below which features are deemed to have low variance. By concentrating computing power on the most informative features, this approach enhances the readability and performance of prediction algorithms. 'Age' and 'BMI' were identified as the selected features in the variance threshold feature selection approach. These features demonstrated variability above the designated threshold, signifying their importance in providing valuable data for the modeling assignment. Consequently, 'age' and 'bmi' were retained in the sample, while features with lesser variance were eliminated. This selection procedure aims to expedite the modeling process while also potentially enhancing the interpretability and performance of the model by focusing on the most informative features.

3.2.7 Standardization

A vital step in data analysis and modeling is to standardize the features inside a dataset, which is accomplished with the help of

the StandardScaler preprocessing technique. It functions by scaling each feature individually to have a standard deviation of one and centering the features around their mean. By ensuring that every feature has a mean value of 0 and a standard deviation of 1, StandardScaler promotes uniformity across the dataset. The interpretability and performance of machine learning models are greatly enhanced by this transformation, which offers a consistent framework for feature comparison and analysis.

3.2.8 One Hot Encoding

A data preprocessing method called "one-hot encoding" is used to convert nominal or categorical data into a binary or numerical representation. In the context of data analysis and machine learning, it is quite helpful. It's used when you want to add categorical data to a predictive model or machine learning technique that requires numerical input. Categorical data represents discrete labels or categories.

One-hot encoding is used to convert each category into a binary vector. For each category, a new binary column is created. In this binary representation, each category is represented by a column. A data item is assigned a value of "1" if it fits into a certain category, and a value of "0" if it does not.

In the dataset, 'smoking_status' is a categorical variable with categories 'never smoked', 'Unknown', 'smokes', and 'formerly smoked'. After applying one-hot encoding, separate columns are introduced for each category. Consequently, for a particular observation where the smoking status is 'smokes', the resulting vector is [0, 0, 1, 0].

3.2.9 Imbalance dataset

The dataset is imbalanced, the count of patients having stroke are 164 and patients having no stroke are 4219.To address class imbalance in the classification task, SMOTE (Synthetic Minority Over-sampling Technique), a popular oversampling method, was employed. SMOTE was utilized to balance the distribution of classes in the dataset by creating artificial samples of the minority class. By generating new synthetic instances using interpolation techniques, SMOTE helps mitigate the effects of class imbalance and enhances the performance of classification models. This approach ensures that the models are not biased towards the majority class and can effectively learn from the minority class instances. The application of SMOTE aimed to improve the robustness and reliability of the classification results, particularly in scenarios with imbalanced class distributions. The number of observations after applying SMOTE is 8438 (4219 from the majority class and 4219 from the minority class).

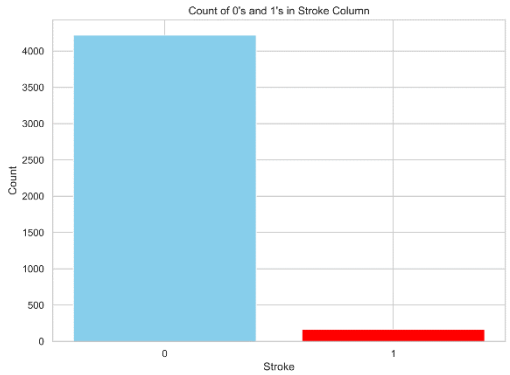


Fig. 10: Count of Stroke and Un-stroke

3.3 Gaussian Naive Bayes

The probabilistic machine learning method Gaussian Naive Bayes which operates well with continuous or real-valued features in classification applications. It relies on the "naive" assumption that, given the class label, all features are conditionally independent, making it particularly suitable for high-dimensional datasets and simplifying the modeling process. A key idea in probability theory, the Bayes theorem, forms the foundation of the method. The "Gaussian" assumption in its name refers to the assumption that the distribution of continuous features is Gaussian (normal), implying that the data are symmetrically distributed around the mean and have a known variance. This distinguishes it from basic Naive Bayes, which can handle categorical data.

The algorithm calculates this probability using Bayes' theorem, wherein the joint probability is expressed as the product of conditional probabilities based on feature independence assumptions. The algorithm determines which class has the highest probability among a collection of inputs by calculating the probability of each class. This allows for the creation of a classifier model. The key parameters to estimate are class probabilities and conditional probabilities, which can be derived from the training data.

3.4 Random Forest

Random Forest stands out as a widely-applied and versatile machine learning method renowned for its proficiency in both classification and regression tasks. This algorithm belongs to the ensemble learning category, skillfully merging the strengths of multiple decision trees to bolster predictive accuracy. In the Random Forest framework, an assortment of decision trees is constructed, with each tree trained on distinct subsets of the training data and a randomly selected subset of features at every node. During the prediction phase, the algorithm aggregates the individual trees' predictions to yield a final outcome. This ensemble strategy effectively mitigates overfitting concerns and enhances the model's ability to generalize to unseen data. Notably, Random Forests exhibit resilience to noisy data and outliers, while also offering interpretable feature importance scores.

3.3 KNN

The K-Nearest Neighbors (KNN) algorithm is a well-liked technique in supervised machine learning that may be applied to both regression and classification issues. KNN uses the proximity principle to predict a new data point's label or value by looking at the labels or values of its K nearest neighbors in the training dataset. Due to its non-parametric nature—that is, the fact that it makes no underlying assumptions about the distribution of the data—it finds use in various domains, including pattern recognition, intrusion detection and data mining. KNN demonstrates its usefulness in practical situations by adapting to both numerical and categorical data, which makes it appropriate for a range of datasets. The method finds the K nearest neighbors based on a chosen distance metric, like the Euclidean distance, and then uses either majority voting for classification or averaging for regression to generate predictions. Notwithstanding its ease of implementation and versatility, KNN presents certain obstacles. Due to its high computational demands and large memory requirements, the approach is not as scalable and is more likely to overfit, particularly in high-dimensional fields. Despite these disadvantages, KNN's predictive potential can be effectively utilized by employing data preparation techniques and careful parameter selection to assist overcome these limitations.

3.4 Hyperparameter Tuning

To improve the model performance, Random Forest hyperparameter tuning entails changing important settings. The efficacy of the algorithm is determined by parameters such as n_estimators, max_depth, min_samples_split, min_samples_leaf, and max_features, which control its behavior. These parameters are systematically changed using methods like grid search and random search, and the model is trained and assessed to determine the best combination. Hyperparameter tuning increases robustness to a variety of datasets, improves generalization, reduces overfitting, and increases model accuracy. It also maximizes computing efficiency, which improves the effectiveness and adaptability of Random Forest models in practical applications.

3.5 K Fold Cross Validation for KNN

i. Data Splitting: K equal-sized folds are created from the dataset.
ii. Training and Validation: Each iteration of the k-NN model is evaluated on the remaining folds after it has been trained on k-1 folds.
iii. Model Training and Evaluation: The k-NN model is trained and evaluated using a specific k value for each fold, taking into account performance metrics such as
accuracy, precision, recall, and F1-score.
iv. Average Performance: To get a general idea of the model's performance for a particular k value, performance data from each fold are averaged.
v. Choosing the Optimal K: Several values for k are assessed, and the value that produces the best performance statistic is selected.
vi. Final Model Selection: The final k-NN model is trained on the selected k value for the entire dataset.

4. RESULTS AND DISCUSSION

The dataset used had 5110 observations and 10 predictor variables, along with a target variable. After preprocessing, the observations were reduced to 4383, while the columns remained 10. One-hot encoding was performed for the categorical variables, as the algorithms used work on numerical variables. Additionally, normalization and standardization were conducted. The binary numerical column 'stroke' was converted into categorical values 'stroke' and 'no_stroke'. Since the dataset was highly imbalanced favoring 'no_stroke', we applied SMOTE (Synthetic Minority Over-sampling Technique), which provided us with a total of 8438 observations.

In this work, three algorithms were used: KNN, Random Forest, and Gaussian Naive Bayes. 25% of the dataset was used for testing, while the remaining 75% was used to train the models. The GridSearchCV methodology, which takes as inputs an estimator, a set of hyperparameters to search over, and a scoring mechanism, was used to automate hyperparameter tweaking. The optimal set of hyperparameters that optimizes the scoring system is returned. This technique, which is incorporated into the scikit-learn toolkit, uses k-fold cross-validation to assess the effectiveness of various sets of hyperparameters. K-fold cross-validation was used for KNN; k = 8 was the optimal number.

Random Forest and KNN had accuracy rates of 93% and 90%, respectively, prior to hyperparameter tweaking. Following the adjustment, the accuracy rose to 94.6% and 96.4%, respectively (optimal k=8). Gaussian Naive Bayes was shown to have an accuracy of 66%. As a result, out of all the models examined, KNN was shown to have the highest accuracy.

In Fig.11, the graph is showing the average cross-validation accuracy for various values of k. Plotting revealed that k=8 produced the best result in terms of bias and variance balancing.

In Fig.12, it demonstrated that increasing the number of trees generally improved model performance up to a certain point, beyond which gains were marginal.

Table 5. Evaluation of models using Statistical Metrics

Model	Precision	Recall	F1_Score	Accuracy	AUC
Random Forest	0.923698	0.974432	0.948387	0.946919	0.989104
KNN	0.949754	0.980114	0.910290	0.964371	0.957920
GaussianNB	0.600459	0.990530	0.747677	0.665403	0.829702

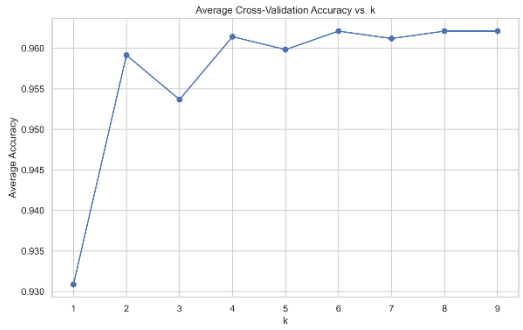


Fig. 11: Avg. cross validation accuracy vs k for KNN

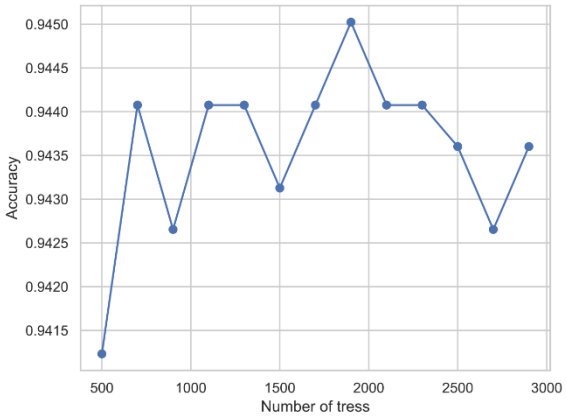


Fig. 12: Avg. cross validation accuracy vs number of trees for Random Forest

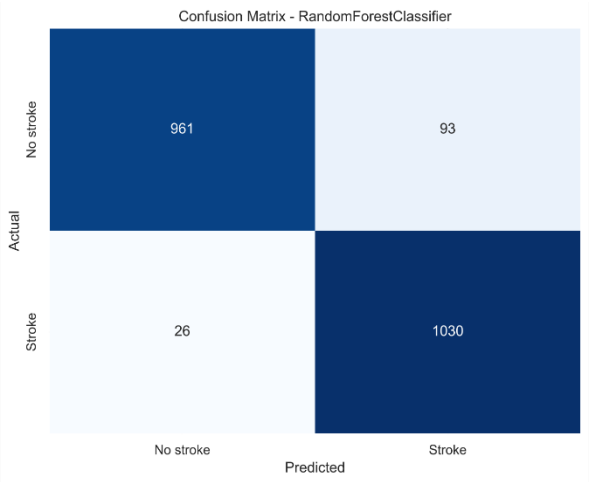


Fig.13: Confusion Matrix For Random Forest

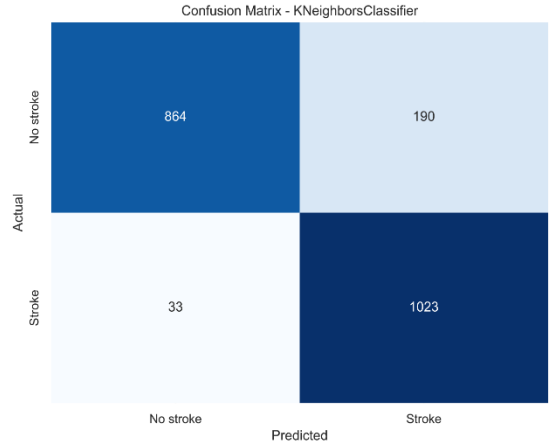


Fig. 14: Confusion Matrix For KNN

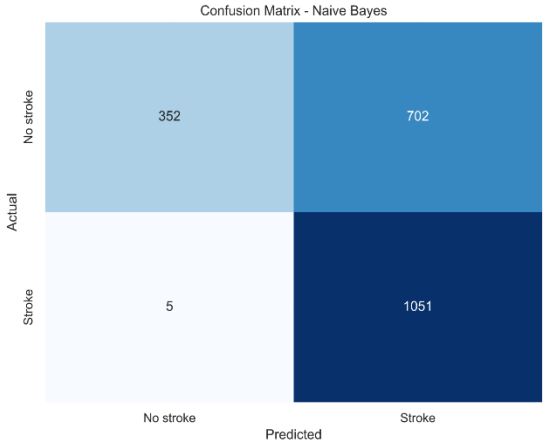


Fig. 15: Confusion Matrix for GuassianNB

5. CONCLUSION

In this study, we investigated how well the machine learning models K-Nearest Neighbors (KNN), Random Forest, and Gaussian Naive Bayes predicted the risk of heart attacks. With careful feature selection, thorough preprocessing, and hyperparameter adjustment, we were able to determine that the KNN model had the highest prediction accuracy of 96.4%. These results highlight how machine learning models can significantly improve risk assessment and intervention tactics in the healthcare industry, leading to better early identification and tailored patient treatment.

The suggested models' robustness and dependability are demonstrated by applying the Synthetic Minority Over-sampling Technique (SMOTE) to solve class imbalance and by thoroughly evaluating them using measures such as precision, recall, F1-score, and AUC. More precise risk assessments may result from the successful application of machine learning in healthcare settings, enabling prompt and efficient medical interventions.

In order to improve accessibility and scalability in clinical settings, predictive model integration with Electronic Health Records (EHRs) should be the main focus of future research. This will automate risk assessment processes. Individual patients could have their machine learning models customized to their specific health profiles and risk factors.

To enhance the accuracy and robustness of the model, more predictive variables such as genetics, lifestyle decisions, and comprehensive medical histories can be incorporated. Additionally, multi-modal data fusion from many sources including imaging investigations and lab findings can be utilized. To guarantee these models' generalizability and practicality, longitudinal research monitoring their efficacy over time and validation in various groups are crucial.

Machine learning has the potential to completely transform risk assessment and intervention techniques in the healthcare industry, resulting in more efficient and individualized patient care.

6. REFERENCES

[1] Majid Khan, Ghassan Husnain, Waqas Ahmad1, Zain Shaukat1, Latif Jan, Ehtesham Ul Haq, Shahab Ul Islam, Atif Ishtiaq: Performance Evaluation of Machine Learning Models to Predict Heart Attack, (2023).

[2] V Sai Krishna Reddy, P Meghana1, N V Subba Reddy, B Ashwath Rao: Prediction on Cardiovascular disease using Decision tree and Naïve Bayes classifiers. Accessed via:DOI:10.1088/1742-6596/2161/1/012015, (2022).

[3] K. Vembandasamy, R R. Sasipriya and E. Deepa: Heart Diseases Detection Using Naive Bayes Algorithm, (2015).

[4] Tanvi S. Patel, Daxesh P. Patel, Mallika Sanyal, Pranav S. Shrivastav: Prediction of Heart Disease and Survivability using Support Vector Machine and Naive Bayes Algorithm. Accessed via: https://doi.org/10.1101/2023.06.09.543776 ,(2023).

[5] Madhumita Pal and Smita Parija: Prediction of Heart Diseases using Random Forest. Accessed via: doi:10.1088/1742-6596/1817/1/012009, (2020).

[6] Xinyi Liu, Siyuan Su, Baoyi Wang, Xuewei Zhang: Prediction of Heart Disease Based on Logistic Regression and Random Forest Models, (2023).

[7] Rupali Atul Mahajan1, Dr. Balasaheb Balkhande, Dr. Kirti Wanjale, Dr. Abhijit Chitre, Tushar Ankush Jadhav, Dr. Sheela Naren Hundekari: Enhancing Heart Disease Risk Prediction Accuracy through Ensemble Classification Techniques. Accessed via: https://www.researchgate.net/publication/375342579, (2023).

[8] Ketut Agung Enriko, Muhammad Suryanegara, Dadang Gunawan: Heart Disease Prediction System using k-Nearest Neighbor Algorithm with Simplified Patient's Health Parameters. Accessed via: t:

https://www.researchgate.net/publication/313717803 ,(2016)

[9] Harshit Jindal: Heart Disease Prediction Using Machine Learning Techniques, (2021).

[10] Shinde, P., Yenkikar, A., Kembhavi, S., Patil, D. (2023). Performance Analysis of Classification Techniques in Heart Disease Prediction. In: Choudrie, J., Mahalle, P.N., Perumal, T., Joshi, A. (eds) IOT with Smart Systems. ICTIS 2023. Lecture Notes in Networks and Systems, vol 720. Springer, Singapore. https://doi.org/10.1007/978-981-99-3761-5_3 (2023).