**Predicting Myocardial Infarction Complications Using Machine Learning**

Author: Harith Palakalmavilavalapil

**Abstract**

Myocardial infarction, commonly referred to as a heart attack, is a severe condition caused by inadequate blood flow to the heart and is a leading cause of death worldwide, affecting nearly 3 million people globally (Mechanic et al., 2023). Prompt and appropriate action post-hospitalization is crucial to saving patients' lives. Clinicians must be extremely vigilant in handling such cases. A decision-support system that predicts the risk for Myocardial infarction (MI) patients can be highly beneficial for clinicians. In this study, we develop a system to anticipate complications in MI patients shortly after hospitalization

The dataset comprises 1700 patient records and 124 attributes, including the target column. The target is a multiclass column with eight distinct classes that indicate whether patients are alive or deceased, along with the cause of death. This dataset is highly imbalanced: 84.06% of the data belongs to class 0, while the remaining 15.96% is distributed among the other seven classes. To address this imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied. Additionally, the dataset contained numerous missing values, which were addressed during data preparation and cleaning.

After data preprocessing applied advanced machine learning methods, including Random Forests, Support Vector Machines (SVM), Extreme Gradient Boosting (XGBoost), and Logistic Regression, to identify the best model for our multiclass imbalanced dataset. To optimize model performance, we conducted hyperparameter tuning using 5-fold cross-validation.

Among the models, SVM emerged as the top performer, achieving an impressive accuracy of 0.995, with both precision and F1 score at a perfect 1. This indicates exceptional classification accuracy and balance between precision and recall, crucial for imbalanced datasets. XGBoost also demonstrated strong performance, with an accuracy of 0.992 and a precision and F1 score of 0.99. Similarly, Random Forest achieved an accuracy of 0.990, with precision and F1 score both at 0.99. In contrast, other models like Decision Tree, K-Nearest Neighbors (KNN), Logistic Regression, and Stochastic Gradient Descent (SGD) underperformed, showing lower accuracy and precision/F1 scores. These results suggest that these algorithms were less effective for this dataset.

In summary, SVM proved to be the most effective model, followed closely by XGBoost and Random Forest. Hyperparameter tuning and cross-validation were crucial in optimizing model performance, ensuring robust and reliable results.

# **Table of Contents**

[Abstract](#_TOC_250033) .2

[Table of Contents](#_TOC_250032) 3

[Acknowledgements](#_TOC_250031) 6

1. [Introduction](#_TOC_250030) 7
   1. [Background to the Project](#_TOC_250029) 8
   2. [Project Objectives](#_TOC_250028) 9
   3. [Report Structure](#_TOC_250027) 9
2. [Literature Review](#_TOC_250026) 12
3. [Requirements](#_TOC_250025) 15
   1. Data set selection 16
   2. System Requirements 16
4. [Methodology 1](#_TOC_250024)6
   1. [Project structure 1](#_TOC_250023)7
   2. [Data Collection 1](#_TOC_250022)8
   3. [Data loading and EDA](#_TOC_250021) 19
   4. [Data Preparation](#_TOC_250020) 19

4.4.1 Handling missing value 21

4.4.2 Data Transformation 21

4.4.3 Handling imbalance data 22

4.4.4 Principal component analysis 23

1. [Train test split](#_TOC_250018) 24
   1. Hyperparameter tuning 24
   2. Bagging 25
2. [Model Development and evaluation](#_TOC_250017) 25
   1. K-Nearest Neighbours (KNN) 26
   2. Decision Tree 28
   3. Stochastic Gradient Descent (SGD): 30
   4. Logistic Regression 33
   5. Random Forest 36
   6. XGBoost classifier 39
   7. Support Vector Machine (SVM): 42
3. [Model comparison 4](#_TOC_250016)5
   1. [Accuracy, precision, recall,F1 score comparison 4](#_TOC_250015)5
   2. [Comparison of AUC of ROC 4](#_TOC_250014)6
   3. [Confusion matrix analysis](#_TOC_250013) 46
4. [Result Conclusion](#_TOC_250010) 47
5. **Ethical Implications of Results 48**
6. [Project Management](#_TOC_250001) 48
   1. [Project Schedule](#_TOC_250000) 48
   2. **Risk Management 49**
   3. **Quality Management 49**
   4. **Social, Legal, Ethical and Professional Considerations 49**
7. **Discussion 50**
8. **Conclusions 51**
   1. **Achievements 51**
   2. **Future Work 51**

**Bibliography and References 53**

**Appendix A – Data set over view 57**

# **Table of Tables**

**Table 1 System requirement 16**

**Table 2 Column deleted from dataset 19**

**Table 3 Classification report of K-Nearest Neighbours (KNN) 26**

**Table 4 Classification report of Decision Tree 29**

**Table 5 Classification report of Stochastic Gradient Descent (SGD) 31**

**Table 6 Classification report of Logistic Regression 34**

**Table 7 Classification report of Random forest 37**

**Table 8 Classification report of XGBoost 39**

**Table 9 Classification report of Support Vector Machine (SVM) 42**

**Table 10 Model comparison 45**

**Table 11 Risk management 49**

# **Table of Figures**

**Figure 1 project structure …………………………………………………………….17**

**Figure 2: missing values before imputation 21**

**Figure 3: missing values after imputation 21**

**Figure 4 Target values before SMOTE 23**

**Figure 5 Target values after SMOTE 23**

**Figure 6 ROC curve of KNN 27**

**Figure 7 Confusion Matrix (KNN) 28**

**Figure 8 ROC curve of Decision Tree 29**

**Figure 9 Confusion Matrix (Decision Tree) 30**

**Figure 10 ROC curve of SGD 32**

**Figure 11 Confusion Matrix (SGD) 33**

**Figure 12 ROC curve of Logistic Regression 34**

**Figure 13 Confusion Matrix (Logistic Regression) 35**

**Figure 14 ROC curve of Random forest 37**

**Figure 15 Confusion Matrix (Random Forest) 38**

**Figure 16 ROC curve of Random Forest (XGB) 40**

**Figure 17 Confusion Matrix (XBG) 41**

**Figure 18 ROC curve of Random Forest (SVM) 43**

**Figure 19 Confusion Matrix (SVM) 44**

**Acknowledgments**

I want to express my deep appreciation to Dr. Ian Dunwell, my supervisor, for his consistent support and guidance throughout my research. His feedback at every stage of the project has been invaluable, and he has always promptly reviewed my progress.

I also want to thank my dear wife for her unwavering support and patience during my studies, always encouraging me to strive for excellence. Furthermore, I am grateful to every faculty member who has instructed me during this course. Their knowledge and guidance have played a crucial role in completing this project.

1. **Introduction**

Myocardial infarction (MI) represents a critical condition where the blood supply to the heart muscle is severely restricted or obstructed, potentially leading to damage or death of heart cells. MI can lead to various complications that may impair heart function and affect other organs, such as pulmonary edema, heart failure, cardiogenic shock, myocardial rupture, and ventricular fibrillation. These complications could significantly increase the risk of death, disability, and hospitalization for MI patients.

Diagnosing MI complications involves considering multiple factors, such as clinical symptoms, electrocardiograms (ECGs), blood tests, various imaging techniques, and the patient's medical history. Currently, diagnosing MI complications relies heavily on the expertise and experience of medical professionals, which can vary among individuals and healthcare institutions. Additionally, human diagnosis may be subject to various influences, including fatigue, biases, and workload.

Machine Learning (ML) leverages extensive data from various sources to uncover intricate patterns and relationships that might be hidden or challenging for humans to discern. Moreover, ML could offer consistent and objective results unaffected by human factors. Furthermore, ML may serve as a valuable tool for healthcare professionals, augmenting their decision-making processes with additional insights, such as predictions, probabilities, explanations, and recommendations.

Given the inherent rarity of certain complications and an abundance of data representing more common outcomes, this dataset exhibited a skewed distribution that could potentially compromise the effectiveness of our machine learning models. To rectify this imbalance, we implemented SMOTE (Synthetic Minority Over-sampling Technique), a statistical approach that achieves a balanced increase in the number of cases within a dataset by creating new instances from existing minority cases.

In this study performed Principal Component Analysis (PCA) by retaining 95% of the variance after standardizing the data for feature selection. This allowed us to reduce the dimensionality of our dataset. To obtain the best machine learning model, I employed random forest classification, XGBoost classifier, SVM, and logistic regression. With the help of the confusion matrix, AUC of ROC, F1 score, and accuracy, all models have been validated.

This study investigates the use of Machine Learning to identify Myocardial Infarction (MI) complications, labelling MI patients based on the presence or absence of specific issues. Multiple Machine Learning models are compared using a wide range of performance metrics.

* 1. **Back ground to the project**

A heart attack, also known as a myocardial infarction (MI), happens when blood supply to a specific portion of the heart is blocked, resulting in tissue damage or death. This blockage is often caused by the accumulation of plaque, a mixture of fat, cholesterol, and other substances, in one or more of the coronary arteries. This plaque can burst, forming a blood clot that totally clogs the artery and causes a heart attack. The heart muscle demands an uninterrupted supply of oxygenated blood to function effectively. When the flow of blood is reduced during a heart attack, the afflicted part of the heart muscle begins to degenerate. Immediate medical care is essential for normalising blood flow, preventing cardiac muscle damage, and retaining heart function.

Heart attacks are a leading cause of death globally. Quick treatment dramatically lowers the risk of death, whereas early intervention increases the chances of survival. Heart attacks, if not treated promptly and properly, can result in serious problems such as heart failure, abnormal heartbeats (arrhythmias), cardiogenic shock, and cardiac arrest. Early intervention can help patients avoid problems and enhance their quality of life. The mortality and morbidity rates linked with heart attack-related heart failure are substantial, emphasising the significance of early detection and treatment. However, even experienced doctors may struggle to accurately forecast heart attack-related problems quickly after admission. Most difficulties occur within the first few hours of hospital admission, emphasising the essential need for prompt and timely management.

To address these challenges and potential delays, the healthcare industry is increasingly turning to artificial intelligence (AI). Early detection of heart attacks is crucial, and machine learning models can significantly improve treatment and early detection. By using comprehensive clinical data, machine learning models can be trained to predict heart attacks more accurately. To address these challenges and potential delays, the healthcare industry is increasingly turning to artificial intelligence (AI). Early detection of heart attacks is crucial, and machine learning models can significantly improve treatment and early detection. By using comprehensive clinical data, machine learning models can be trained to predict heart attack outcomes, enabling prompt and accurate treatment.

* 1. **Project Objectives**

The primary objective of this research project is to evaluate the viability of using Machine Learning (ML) techniques to predict complications related to myocardial infarction (MI). This study will address two central research questions:

1. Which machine learning technique is most accurate in predicting complications arising from myocardial infarction?
2. Can machine learning effectively handle a multiclass imbalanced dataset related to myocardial infarction?
   1. **Report Structure**

This report is divided into 12 main sections with each section focus on a specific set of activities as described below. Additional information and materials are provided in Appendix for reference.

Section 1: This section provides introduction, background of the project and project objectives.

Section 2: This section mainly covers literature review to highlight work of other researchers in this field and helps to understand current progress in this field of research.

Section 3: This section covers requirements that’s required to complete the project. Section 2 gives the idea about the suitable data set that is required for my project.

Section 4:This section outlines the methodology employed in this project, detailing each step taken to prepare and analyse the data.

* **Overview of the Input Dataset**: This subsection provides a comprehensive overview of the dataset, highlighting all its attributes and their significance.
* **Exploratory Data Analysis (EDA)**: Various plots and visualizations are created to understand the features and their influence on the target variable. This analysis aids in identifying trends, patterns, and relationships within the data.
* **Data Cleaning**: Through the EDA process, columns that do not significantly impact the target variable are identified and subsequently removed from the dataset to enhance model performance.
* **Data Preparation**: A Label Encoder is applied to convert categorical values into numerical representations, making them suitable for machine learning algorithms.
* **Handling Missing Values**: Missing data is addressed using KNNImputer, which estimates and fills in missing values based on the nearest neighbours’ values.
* **Z-Score Normalization**: This technique is utilized to standardize numerical features, ensuring that they all have a mean of zero and a standard deviation of one, facilitating better model convergence.
* **Addressing Data Imbalance**: The Synthetic Minority Over-sampling Technique (SMOTE) is employed to manage class imbalance in the dataset, enhancing the model's ability to learn from minority classes.
* **Feature Selection**: Principal Component Analysis (PCA) is implemented to reduce dimensionality and select the most significant features, improving model efficiency and interpretability.

Section 5: This section deals with the implementation of test and train split of data set after the data preprocess techniques applied in section 4. In this section to avoid overfitting hyper tuning technique and bagging technique has been used.

Section 6: Following section 5 this section explains different machine learning techniques that applied on this research. This section also discusses the classification reports, ROC curve, Confusion matrix obtained from each model.

* Random forest
* XGB classifier
* Support vector machine (SVM)
* Logistic Regression
* K-Nearest Neighbours (KNN)
* Stochastic Gradient Descent (SGD)
* Decision Tree

Section 7: This section covers evaluation of the results that obtained from section 6

* Accuracy, precision, recall, F1 score comparison
* Comparison of AUC of Roc curve
* Comparison of confusion matrix

Section 8: This section concludes project finding and provide final results.

Section 9: This section addresses the ethical implications related to the project's results

Section 10: This section covers project management aspects of this research and briefly explains key processes.

Section11: This section provides student reflections.

1. **Literature Review**

The study, "Machine Learning Prediction of Mortality in Acute Myocardial Infarction" (Oliveira et al., 2023), investigates the application of machine learning (ML) models to predict in-hospital mortality in patients with acute myocardial infarction (AMI). This literature review aims to identify common methods used for predicting myocardial infarction with machine learning. Traditional approaches, such as the TIMI score, often fall short in terms of accuracy and predictive power. The study examines ML models including Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Stochastic Gradient Descent. To improve model performance, the research employs feature selection techniques and oversampling methods to address data imbalance, a frequent issue in medical datasets.

Among the tested ML models, K-Nearest Neighbours (KNN) with oversampling demonstrated the best performance, achieving an Area Under the Receiver Operating Characteristic Curve (AUC) of 89% and a recall rate of 90%. These results represent a significant improvement in predictive accuracy compared to traditional methods. The research underscores the importance of integrating comprehensive clinical and laboratory data into ML models to capture complex patterns and associations within patient data, leading to better predictions for acute myocardial infarction (AMI) outcomes. The findings suggest that ML models, when combined with techniques to address data imbalance and supplemented with complete patient data, offer a promising approach for enhancing in-hospital mortality predictions for AMI patients. This improvement could potentially enable more timely and effective clinical interventions. The study highlights the potential of ML models, particularly those utilizing holistic patient data, to enhance the accuracy of predictions and patient outcomes.

A study of Prediction of Myocardial Infarction From Patient Features With Machine Learning by Chen et al., 2022 looked into using machine learning models to automatically assess the severity of myocardial infarction (MI) in patients. They create models to assess MI severity using readily available data (physiological, clinical, and paraclinical aspects) obtained during emergency department visits. Experiments were carried out on 150 examples and evaluated using cross-validation. Two models were utilised to predict the outcome. In the classification model, categorise individuals into groups based on the presence or absence of MI and persistent microvascular obstruction (PMO), a complication of MI. The regression model predicts the percentage of infarcted myocardium (PIM), which quantifies the amount of the injury. The data set contains information on 150 patients suspected of having an acute MI. The truth labels (real severity) were obtained from delayed enhancement MRI DE-MRI examinations and manual annotations. Cross-validation was used to assess the model's performance. In this study, they employed Support Vector Machines (SVMs), Random Forests (RF), Decision Trees, and boosting techniques such as Gradient Boosting Decision Trees to manage nonlinear relationships. The results showed that for the MI (PMO inclusive) and the PMO (infarct exclusive), the best models acquired respectively a mean error of 0.056 and 0.012 for quantification, and 88.67 and 77.33% for classification accuracy of the myocardial state.

Another study by Chumachenko et al., 2022 aims to analyse 10-second electrocardiogram (ECG) measurements taken from 12 leads to classify patients with suspected myocardial infarction (MI) using machine learning methods. Four models were developed for this purpose: k-nearest neighbour (KNN), radial basis function (RBF), decision tree, and random forest. The experimental investigation was conducted using the open PTB-XL dataset, which contains data from patients with suspected MI. The results demonstrated that short ECG parameters could effectively classify patients with suspected MI as either sick or healthy with high accuracy. Among the models, the optimized random forest model performed the best, achieving an accuracy of 99.63% and a root mean absolute error of less than 0.004. This novel approach can be particularly useful for diagnosing patients who do not exhibit other typical indicators of heart attacks. Author saying that automated diagnostics using machine learning are essential for the digitalization of healthcare and the advancement of personalized medicine, including applications in smart homes. In order to achieve this, For ECG diagnostics, early data collection potentially by patients or their relatives at home is crucial. While a traditional 12-lead ECG may not be practical for home use, a wearable patch enabling heart rate variability (HRV) analysis supported by AI is suggested. Overall, this work indicates how machine learning can improve the efficiency and accuracy of MI severity evaluation in clinical settings.

Divneet Mandair et al., 2020 examines the application of machine learning (ML) to predict myocardial infarction (MI) using harmonized electronic health record (EHR) data. Traditional risk prediction methods like the TIMI Risk Score have been commonly used but rely on established risk factors and have not significantly advanced with new technology​​. Recent studies have focused on using ML and deep learning (DL) to improve MI risk prediction. This study evaluates different ML models, such as deep neural networks (DNNs), random forest, and gradient boosting machines, against traditional logistic regression. The dataset includes 2.27 million patients and over 52,000 features, standardized to the Observational Medical Outcomes Partnership common data model (OMOP-CDM). The goal is to predict MI occurrence within six months​​. The study discovered that a deep neural network (DNN) with random under sampling performed better than other models, although a logistic model based on known MI risk variables was nearly as accurate. All models had poor calibration. While ML approaches, notably DNNs, provided little advantage over traditional tools, the study's methodology and use of harmonized data were important steps towards establishing scalable prediction tools for clinical practice. This indicates that although ML techniques have potential, they need further development and validation to be more effective than conventional approaches in clinical practice. Hence, this dissertation seeks to provide such further development and validation by refining the machine learning models, optimizing data integration techniques, and conducting comprehensive evaluations to enhance their performance and applicability in real-world clinical settings.

The study "Risk Prediction of Heart Failure in Patients with Ischemic Heart Disease Using Network Analytics and Stacking Ensemble Learning" (Li et al., 2021) explores the potential of machine learning in predicting heart failure post-myocardial infarction. Published in BMJ Open, this research underscores the growing importance of advanced analytics in clinical settings. The authors leverage network analytics to identify complex interactions among various clinical parameters, subsequently employing stacking ensemble learning to enhance predictive accuracy. This combination aims to overcome the limitations of traditional prediction models, which often fail to capture intricate variable relationships and provide suboptimal accuracy. The study emphasises how effective ensemble approaches are, especially stacking, which combines several machine learning algorithms to maximise performance. Through the use of this methodology, the researchers were able to attain significant gains in prediction accuracy, he XGBoost classifier performed the best with a ROC-AUC score of 0.8416, indicating the usefulness of the model in clinical settings. The improved model offers patients who are at risk of heart failure after myocardial infarction a reliable tool for early intervention and individualised treatment plans. Even with the encouraging outcomes, the study recognises the difficulties in putting such sophisticated models into practice, such as the significant processing overhead and the demand for thorough validation across a range of patient populations. To guarantee dependable model performance, the integration of electronic health records (EHRs) also brings data quality and standardisation challenges that need to be resolved. In conclusion, this research emphasizes the transformative potential of machine learning in cardiology, advocating for further exploration and refinement of these techniques to improve patient outcomes.

Farah et al., 2022 says the application of machine learning (ML) in healthcare has shown promising advancements, particularly in predicting patient outcomes. Prior studies have demonstrated ML's potential in diagnosing cardiovascular diseases and forecasting patient mortality. For instance, ML models have been effectively used to predict heart failure and readmission rates, significantly outperforming traditional statistical methods. In the context of myocardial infarction (MI), research has emphasized the importance of early detection and timely intervention. However, existing models often lack generalizability and robustness due to limited datasets and feature variability. Recent work has started to address these issues by incorporating larger, more diverse datasets and advanced ML techniques such as ensemble learning and deep learning. Despite these advancements, there remains a critical gap in accurately predicting short-term mortality post-MI, highlighting the need for further research to enhance predictive accuracy and clinical applicability.

In this study propose an ML-powered framework to predict the lethal outcome of MI patients. Using a dataset of 1,700 subjects and 111 clinical characteristics, this dissertation trained various ML models. Cox Regression was implemented to study the effect of various clinical phenotypes on the probability of patient survival. Feature selection techniques ,sequential forward floating selection and recursive feature elimination were applied to identify the most relevant features for the models. Among the evaluated models, the logistic regression classifier achieved notable performance with an accuracy of 86.47% and a weighted F1 score of 86.92%. This study seeks to bridge the gap in predictive accuracy and clinical applicability, offering a robust approach to MI outcome prediction.

**3. Requirements:**

This section details the approach taken to finalize the project requirements. Since this project selected the waterfall methodology for software development, as outlined in Section 3, the requirements are as follows:

**3.1 Data set selection:**

I chose a dataset from the UCI Machine Learning Repository, which includes 1,700 records from the Krasnoyarsk Interdistrict Clinical Hospital in Russia. This dataset is designed to analyse and predict complications related to myocardial infarction (MI) in patients, making it suitable for developing a classification machine learning model. It provides real-world clinical data, which is essential for creating models that can be effectively applied in actual healthcare settings. This enhances the practical value of the research.

**3.2 System Requirements:**

**Table\_1: System requirement**

|  |  |
| --- | --- |
| System type | 64-bit operating system, x64-based processor |
| Operating system | Windows 11 Pro, |
| Processor | intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz |
| RAM | 8.00 GB |
| Device ID | B1644B14-BCA7-4765-AB72-261265EBF11F |

**4. Methodology**

The software development lifecycle (SDLC) is an effective and efficient method for designing and constructing reliable software. Its primary aim is to mitigate project risks through proactive planning, ensuring that the software meets customer expectations (AWS, 2023). software development lifecycle (SDLC) enables efficient planning and scheduling of development activities. There are several SDLC models to choose from, and for this project I opted for the Waterfall model. This model organizes project phases in a sequential manner, where each phase relies on the deliverables of the preceding phase. In essence, the progression from one phase to the next resembles the steady flow of a waterfall.

**4.1 Project structure**

**Figure\_1\_project structure**

Data cleaning

Data collection

Missing data handling

Data standardization

Data Loading &EDA

Hyper parameter tuning

Model creation

Training set

Imbalance data handling

Feature selection

Data preparationgTable 4 Topsoil Attributes in dataset

Test set

Train test split

Model evaluation

**4.2 Data collection**

The UCI Machine Learning Repository released a database with 1700 records from Krasnoyarsk Interdistrict Clinical Hospital in Russia, aimed at analysing and predicting complications associated with myocardial infarction (MI) in patients (UCI Machine Learning Repository, n.d.)

. The dataset consists of 124 properties, with 111 of them containing information on patient age, gender, medical history, conditions upon admission, ECG results, and clinical interventions post-admission. The remaining properties document different complications at four specific time points: upon admission, 24 hours post-admission, 48 hours post-admission, and 72 hours post-admission. Moreover, the dataset includes details on causes of death, categorized into seven groups. This dataset comprises binary columns, categorical columns, and integer columns.

This study predicts patient risk upon admission, a critical factor for ensuring patient survival. From the attributes, Lethal outcome (cause) (LET\_IS) is my target column. Which contains the cause of Myocardial Infarction and it's a multiclass features column. This attribute contains 8 classes. Classes are given below with a brief. Briefs are based on (cardiovascular diseases (CVDs), n.d.)

Lethal outcome (cause) (LET\_IS)

0: Unknown (alive): The individual is alive.   
1. Cardiogenic shock: This syndrome occurs when the heart fails to deliver enough blood to meet its requirements and is typically the result of a severe heart attack.   
2. pulmonary oedema, which has been defined as an accumulation of fluid in the lung tissue that makes breathing hard.

3: myocardial rupture - A myopic episode is usually followed by a rupture in the myocardium.

4: progress of congestive heart failure - **Congestive heart failure (CHF)** is a chronic condition that Fluid accumulation results from a weakening of the heart's pumping action over time.

5: thromboembolism -A blood clot breaks loose and blocks a blood vessel.

6: asystole- **Asystole** is a medical term for when the heart's electrical activity completely stops.

7: ventricular fibrillation - **Abrupt, erratic heartbeats that impair the heart's ability to pump blood efficiently.**

**4.3 Data loading and EDA**

The dataset of myocardial infarction (MI) records was loaded into Python memory, and an exploratory data analysis (EDA) was performed. This process provided valuable insights into the data attributes, checked for data integrity, examined data distribution, identified outliers, and generated a visual summary of the dataset. These insights are crucial for the subsequent data preprocessing step. Histograms were used to analyse the distribution of the dataset, allowing us to understand the frequency and spread of various attributes. This helped identify any skewness or anomalies in the data distribution. Relationships between the target variable and the remaining attributes were explored using scatterplots, which helped determine how different factors influence the outcome. Continuous variables were examined for outliers, which can significantly impact the performance of predictive models. It was found that four attributes had major outlier issues, and some columns did not show any relation with the target variables.

To further investigate the relationships between variables, a heatmap was plotted using the Seaborn library (Appendix C). This heatmap visualized the correlation between different attributes, highlighting which variables are strongly correlated and which are independent. Understanding these correlations helps in feature selection and engineering, ensuring that the most relevant variables are included in the predictive model. The heatmap revealed that some columns were not related to any of the other variables.

**4.4 Data Preparation:**

Data preparation is an important stage in any machine learning research. Input data frequently requires multiple modifications before it can be used for model training. This procedure includes data cleaning and modifications such as standard scaling, dimensionality reduction, and translating categorical variables to numeric representations suited for machine learning.

In my analysis, I reviewed the data and removed the following columns that don't have a direct impact on the target variable "LET\_IS" (Lethal outcome (cause)). Removed columns are listed below.

**Table1\_ 2 Column deleted from dataset**

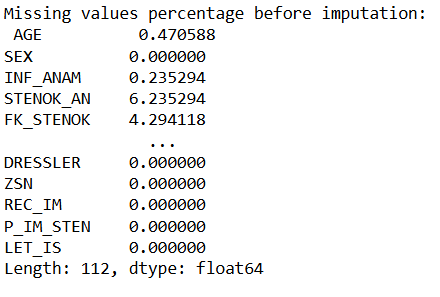
|  |
| --- |
| ID |
| NA\_R\_3\_n |
| R\_AB\_1\_n |
| R\_AB\_2\_n |
| R\_AB\_3\_n |
| NA\_R\_1\_n |
| NA\_R\_2\_n |
| NA\_R\_3\_n |
| NOT\_NA\_1\_n |
| NOT\_NA\_2\_n |
| NOT\_NA\_3\_n |
| NA\_R\_1\_n |
| NA\_R\_2\_n |

Out of the 124 attributes, 111 provide information on patients' demographics, medical history, complications at admission, and clinical data. The remaining 12 columns capture various complications at four specific time points: i) at the time of admission, ii) 24 hours post-admission, iii) 48 hours post-admission, and iv) 72 hours post-admission. From these 12 columns, I retained only the ones related to the time of admission and removed the rest. At present, the dataset comprises 112 features and 1700 rows. There are still a few missing values, approximately 6.25% of the total dataset.

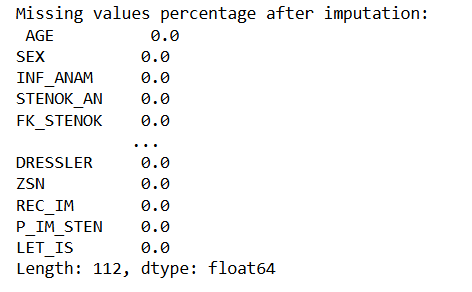
**4.4.1 Handling missing value**

Handling missing values is essential to ensure reliable and accurate results when working with machine learning models. There are various techniques for addressing missing data, such as imputation (replacing missing values with statistical estimates like mean, median, or mode) or removing rows/columns with missing values, depending on the nature and quantity of missing data. In my case, I chose to apply KNNImputer, which is a technique for addressing missing data in datasets. KNNImputer works by guessing missing values using the values of their closest neighbours in the feature space. Essentially, for each missing value, the algorithm determines its k nearest neighbours using a distance metric (such as Euclidean distance) and then computes a weighted average of their values. This method utilizes the local associations between data points to impute missing values, which is especially useful when data is missing at random or in datasets with complex interactions between variables.

**Figure2\_ missing values before imputation**



**Figure\_3 Missing values after imputation**

****

**4.4.2 Data Transformation**

This dataset comprises several categorical columns that must be converted into numerical representations to optimize model performance. Categorical data can be classified into different types, each requiring specific conversion methods. For instance, nominal categorical variables—those without a natural order, such as colours or types of animals—are best transformed using one-hot encoding. This technique creates binary columns for each category, allowing models to interpret the data effectively.

In contrast, our dataset contains ordinal categorical data, which has a specified logical order. Choosing the right encoding method for ordinal data is critical, as the wrong option might distort results and mislead analysis. In this case, label encoding is used to turn categorical variables into numerical values, with each category assigned a unique integer based on its order.

After converting the categorical variables standardize the numerical features using Z-score normalization. This statistical method involves subtracting the mean of each feature from the individual feature values and then dividing by the feature's standard deviation. The result is a dataset where each feature has a mean of zero and a standard deviation of one. Standardization is crucial for several reasons. Firstly, it ensures that all features contribute equally to the distance calculations used in many algorithms, particularly those sensitive to feature scale, such as gradient descent-based methods. This helps algorithms converge more quickly and effectively, as they can better navigate the optimization landscape without being skewed by features with larger ranges. Additionally, standardized data allows for meaningful comparisons between features, as they are now all on a similar scale. Equation (1) refers to formula for calculating standardization.

𝑥ᇱ = 𝑥 − 𝜇 𝜎 (1)

Overall, this process enhances the model's ability to learn from the data, leading to improved predictions and more reliable outcomes. By carefully choosing the right encoding techniques and standardizing the data, this study lay a solid foundation for effective model training and evaluation.

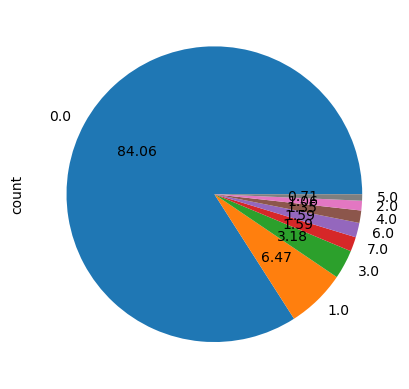
**4.4.3 Handling Imbalanced Data**

The myocardial infarction (MI) dataset is significantly imbalanced, which can hinder the performance of predictive models. Addressing this class imbalance is crucial for improving prediction accuracy. Various methods are available to handle imbalanced data, such as under-sampling, oversampling, SMOTE, and SMOTE-ENN.

For this high-dimensional, multiclass classification task, I chose to use SMOTE (Synthetic Minority Over-sampling Technique). SMOTE is particularly effective because it generates synthetic samples for the minority class by interpolating between existing minority class instances. This helps in balancing the class distribution without simply duplicating existing samples, which can lead to overfitting.

SMOTE is well-suited for high-dimensional data as it can better capture the complex relationships between features compared to simple oversampling techniques. By using SMOTE, the model can learn from a more balanced dataset, leading to improved generalization and more accurate predictions across all classes.

**Figure\_4\_Target values before SMOTE**



**Figure\_5- Target values after SMOTE**

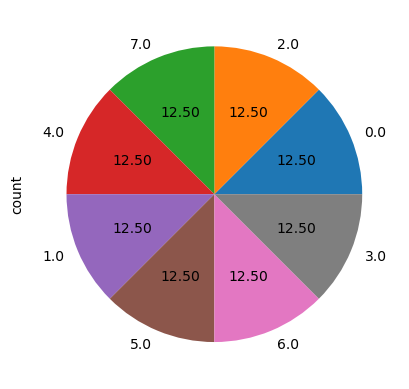


Figure 4 showing the percentage of distribution of target variable before SMOTE operation.

Figure 5 showing persentage of distribution of target variable after SMOTE operation.

**4.4.4 Principal Component Analysis (PCA)**

The dataset initially contained 124 features, which were reduced to 85 features after feature selection. To further enhance the performance of the machine learning model, it was crucial to reduce the number of features using dimensionality reduction techniques. Various methods can achieve this, such as Principal Component Analysis (PCA) and Ridge Regression. For this project, I chose PCA for dimensionality reduction.

PCA is a widely-used and efficient technique that transforms the original features into a set of orthogonal components, capturing the most variance in the data. By retaining only the most significant components, PCA reduces the dimensionality while preserving the essential information, leading to improved model performance and reduced computational cost. Before PCA, the number of features was 112. After applying PCA, the number of features was reduced to 85. This reduction helps in minimizing overfitting, enhances the generalization capability of the model, and speeds up the computation process, making the machine-learning pipeline more efficient and robust.

**5. Train test split**

The processed dataset is split into two parts: training and testing. In this split, 70% of the data is allocated for training the machine learning model, while the remaining 30% is reserved for testing its performance. This division ensures that the model is evaluated on unseen data, providing a more accurate assessment of its predictive capabilities.

The specific split is performed using the train\_test\_split function from scikit-learn, which ensures that the data is divided randomly. However, to ensure that the results are reproducible and consistent across different runs, a random state of 42 is set. This means that every time we run the code with the same random state, ww will get the same training and testing sets, allowing for consistent model evaluation and comparison.

After preprocessing the data and splitting it into training and testing sets, the next step is to build and evaluate various machine learning models. For this project, several classification techniques will be employed, including Support Vector Machines (SVM), Logistic Regression (LR), Random Forest, and XGBoost, K-Nearest Neighbours (KNN), Decision Tree. Given the complexity of the dataset, hyperparameter tuning and advanced techniques like Bagging (Bootstrap Aggregating) will be used to enhance model performance.

* 1. **Hyperparameter tuning:**

Due to the significant imbalance between the target classes, it is essential to perform outlier treatment on this dataset. In the initial stage, without hyperparameter tuning, observed overfitting across all models, resulting in poor performance. To address these issues, this project utilizing the Grid Search approach for hyperparameter tuning. This method evaluates all possible combinations of a given set of hyperparameter values to identify the best configuration for a machine learning model. According to Belete and Huchaiah's 2021 research study, tuning hyperparameters has a statistically significant positive impact on model prediction accuracy. In this project, I used fivefold cross-validation, meaning the dataset is divided into five parts. The model is trained and validated five times, each time using a different portion for validation and the remaining parts for training. The main steps involved in the grid search are outlined below.

* Recognising Hyperparameters:

Finding and adjusting the hyperparameters is the initial stage in the procedure. These variables are pre-set before to training and are unique to the model type being employed. For instance, the minimum number of samples needed to split an internal node (min\_samples\_split), the minimum number of samples needed to produce a leaf node (min\_samples\_leaf), and the maximum depth of the tree (max\_depth) are examples of hyperparameters in a decision tree model. The regularisation parameter (C), the kind of regularisation (penalty), and the optimisation solver (solver) are crucial hyperparameters in logistic regression. The hyperparameters that determine how well a model performs during training are specific to each model.

* Select a Search Strategy:

Two common search strategies are Grid Search and Random Search. In this project, Grid Search is utilized. Grid Search exhaustively evaluates all possible combinations of specified hyperparameter values, ensuring the optimal set is identified. This comprehensive approach guarantees a thorough exploration of the hyperparameter space for the best performance. Additionally, Grid Search integrates cross-validation, where the training data is split into multiple folds and each fold is used as a validation set in turn. This method enhances the assessment of the model's robustness and helps prevent overfitting.

The final model is trained using these ideal settings on the whole training dataset after the best hyperparameters have been determined. This guarantees that all of the data is utilised by the model to boost performance and generalisation and that it is properly optimised.

**5.2 Bagging (Bootstrap Aggregating):**

Bagging, or Bootstrap Aggregating, is a robust ensemble learning method designed to enhance the stability and accuracy of machine learning models. Its primary objective is to reduce variance and improve prediction reliability by combining models trained on different samples of data. As a key technique in ensemble learning, bagging utilizes multiple models to form a more consistent and accurate prediction framework. By training on various subsets of the data and aggregating their results, bagging effectively mitigates issues such as overfitting and high variance.

**6. Model Development and evaluation:**

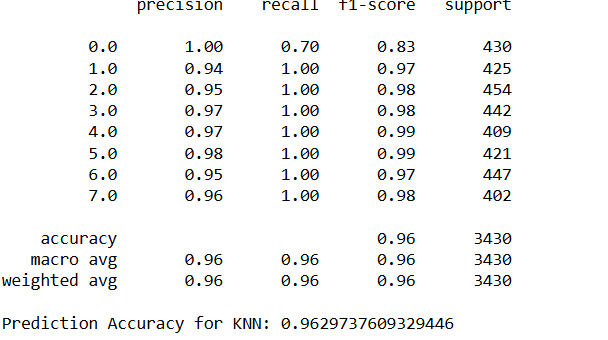
For this project, various classification techniques are employed to identify the most suitable model for the dataset. The methods implemented include Support Vector Machines (SVM), Logistic Regression (LR), Random Forest, and XGBoost. Each model operates on different underlying principles, which results in varying performance metrics.

All models were evaluated using multiple metrics to ensure a comprehensive assessment of performance. The metrics included the AUC of the ROC curve, which measures the model's ability to distinguish between classes, and train-test prediction accuracies, which reflect how well the model generalizes to unseen data. Additionally, the F1 score, recall, and precision were used to balance the trade-off between false positives and false negatives. The confusion matrix provided detailed insight into the model's performance across different classes, helping to identify any biases or issues with imbalanced classes. These combined metrics ensure a robust evaluation of the model's predictive capabilities.

**6.1 K-Nearest Neighbours (KNN):**

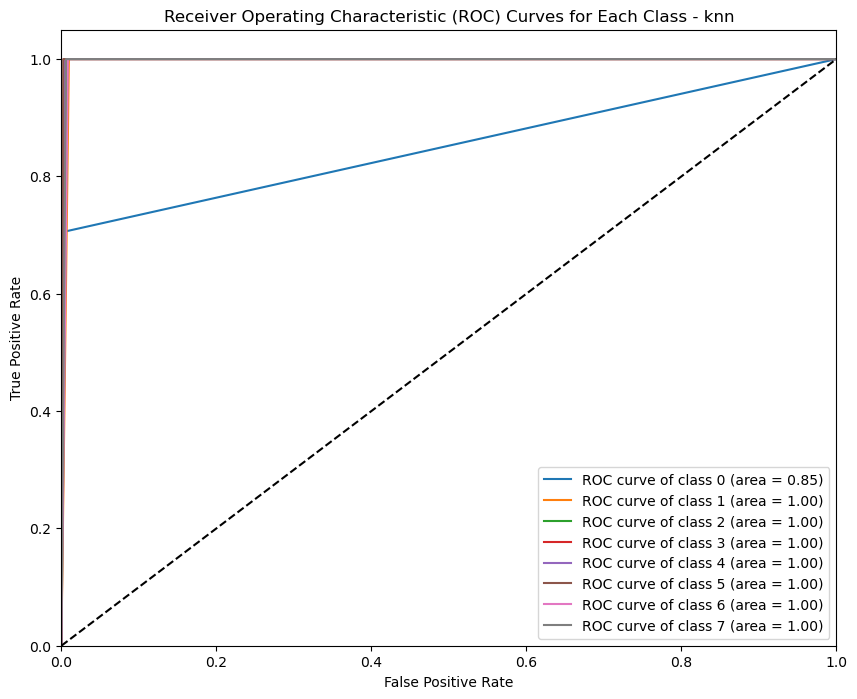
K-Nearest Neighbours (KNN) is a simple, non-parametric technique utilized for both classification and regression. It functions by storing training data and, when making predictions, computing the distance between the new data point and all existing points. Then the algorithm identifies the 𝑘. k nearest neighbours and assigns the most frequent class among them (for classification) Although KNN is straightforward to comprehend and implement, it may be computationally demanding for large datasets and susceptible to irrelevant features and high dimensionality. The choice of the best 𝑘 and distance metric is critical for performance.

**Table\_3: Classification report of K-Nearest Neighbours (KNN):**



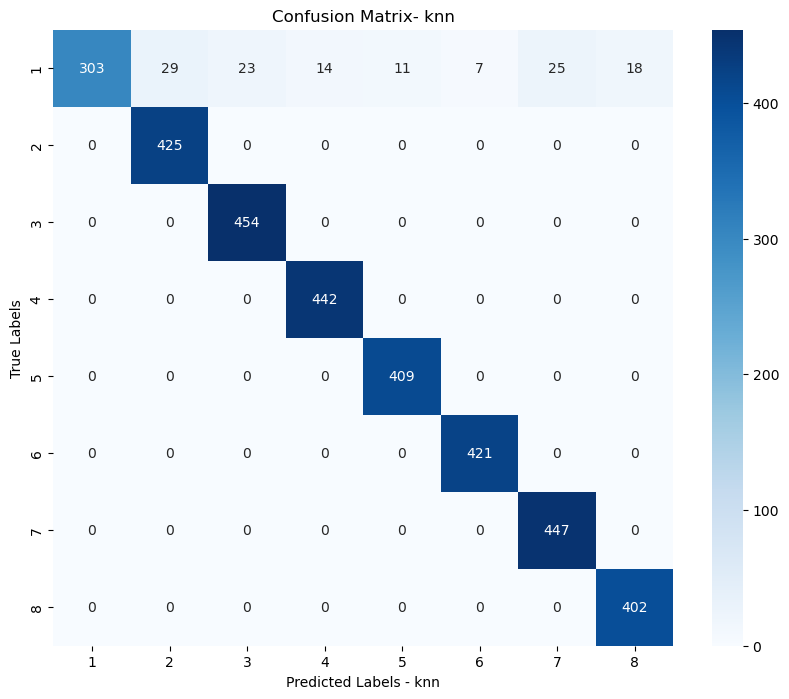
Bagging with KNN exhibits a training accuracy of 97.76% and a testing accuracy of 96.30%, indicating a well-fitted model with minimal overfitting. Overall accuracy is 0.962 The classification report highlights some variation in performance across different classes. Specifically, class 0.0 has a lower recall of 0.70, indicating difficulties in correctly identifying all instances of this class. Despite this, most other classes have high precision, recall, and F1-scores, showcasing the model's effectiveness in general. While the overall accuracy and metrics are strong, the lower recall for class 0.0 suggests that further improvements could be made. Integrating other models could enhance performance, particularly for the underperforming classes. This balanced approach ensures robust and reliable classification across all classes, leading to better overall model performance.

**Figure\_6: ROC curve of KNN**



The KNN classifier performs exceptionally well, with an AUC of 1.00 for classes 2, 3, 4, 5, 6 and 7, indicating perfect classification. For classes 0 the AUCs are slightly lower at 0.85 reflecting slightly weaker but still robust performance. These results suggest that the KNN model is highly effective in distinguishing most classes, achieving strong overall performance. The only minor area for improvement is in class 0, but even there, the classifier maintains a commendable AUC above 0.85, highlighting its reliability across all classes.

**Figure\_7: Confusion Matrix (KNN)**



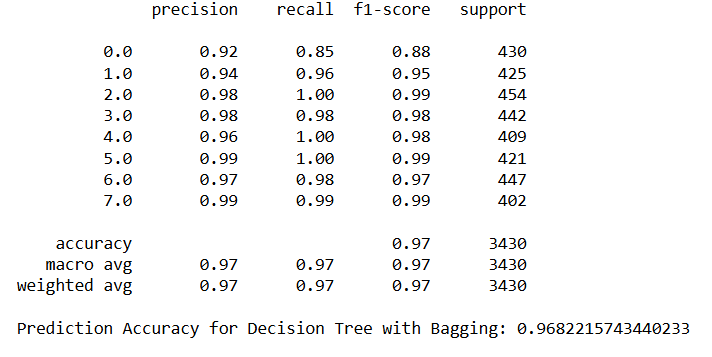
A confusion matrix evaluates a classification model by comparing predicted and true labels. The diagonal cells (top-left to bottom-right) represent correct predictions where true and predicted labels match. Higher values in these diagonal cells indicate better model performance, reflecting more accurate predictions. Maximizing these values is crucial for an effective model.

By analysing this confusion matrix, we observe that the diagonal elements have higher values, indicating a large number of correct predictions. The off-diagonal cells represent incorrect predictions, such as cell (1, 2) with a value of 29, meaning 29 instances of class 1 were misclassified as class 2. Compared to other classes this model misclassified class 1 to other classes. Since there are relatively few numbers in the off-diagonal cells compared to the diagonal elements, in conclusion this model performs moderately well.

**6.2 Decision Tree:**

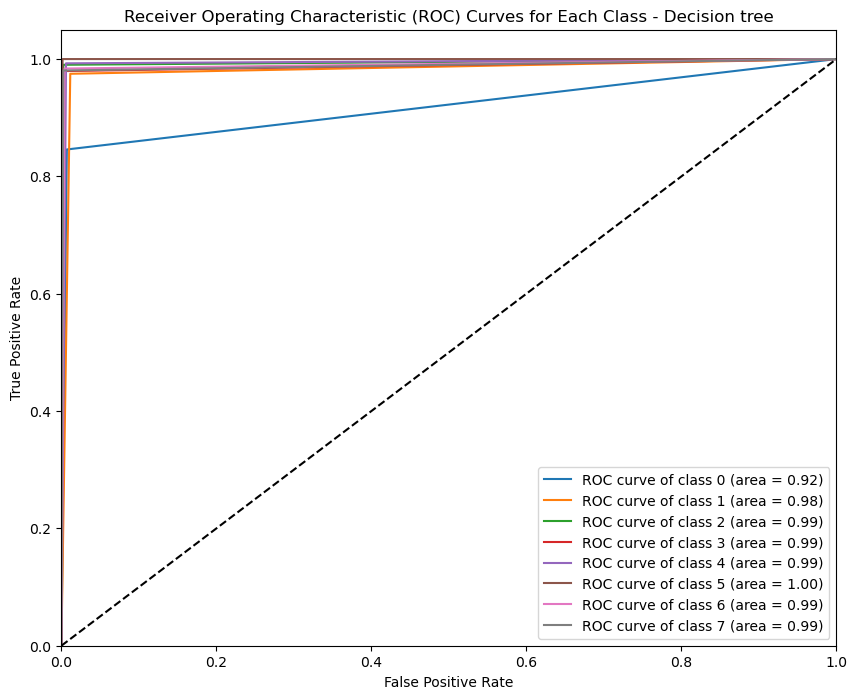
Decision Trees are a versatile supervised learning algorithm for both classification and regression tasks. They operate by recursively splitting data based on feature values, resulting in a tree structure where internal nodes represent decisions, branches denote outcomes, and leaves signify class labels or regression results. This tree-based model is highly interpretable, allowing easy visualization of decision processes. However, the main disadvantage of decision tree is it can easily overfit, especially with deep trees capturing noise in the data. Pruning methods is used to address this issue. Despite these challenges, Decision Trees are effective in capturing non-linear relationships and are often used as base learners in ensemble methods like Random Forests and Gradient Boosting.

**Table\_4: Classification report of Decision Tree**



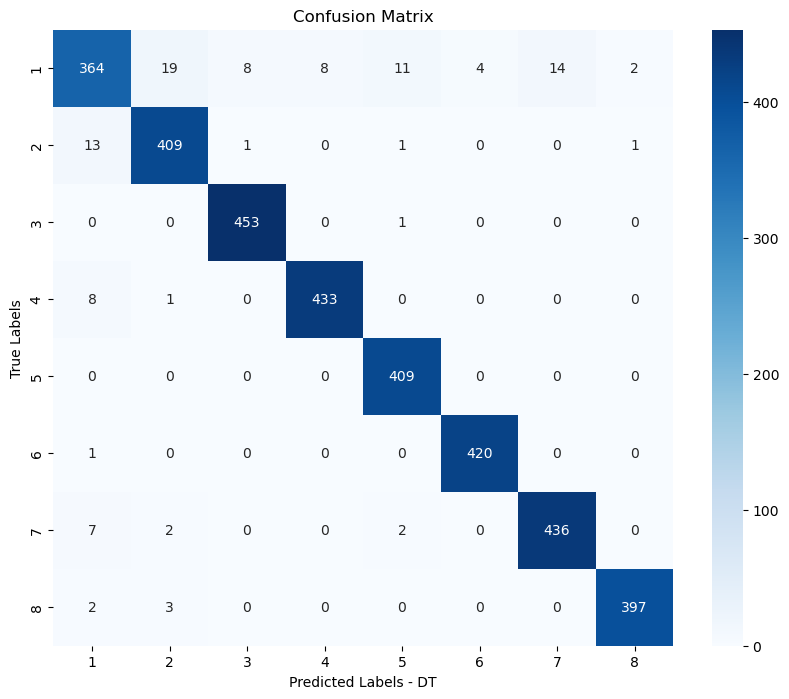
Based on the research findings, the Decision Tree shows average precision, recall, and F1 scores of 0.97, indicating strong performance. The overall predictive accuracy is 0.968. However, there are indications of overfitting as the model has a high training accuracy of 99.98% but a lower testing accuracy of 96.82%. This suggests that the model performs extremely well on training data but struggles to generalize to new data, which is a common issue with overfitting. It's well-known that decision trees are prone to overfitting, especially when they become too large or when the dataset is noisy. Deep decision trees may recall training data, leading to poor generalization to unknown data.

**Figure 8: ROC curve of Decision Tree**



The AUC for class 0 is 0.92, which is comparatively lower than other classes. Classes 2, 3, 4, 6, and 7 have an AUC of 0.99. Notably, class 5 has an AUC of 1, showing perfect performance, suggesting the model excels in these classes. However, class 1 is an exception where the model's performance is not as strong. Overall, the model performs well across most classes, with the highest proficiency in class 5, demonstrating its robustness and effectiveness in handling multiple categories with good accuracy.

**Figure 9: Confusion Matrix - Decision Tree**



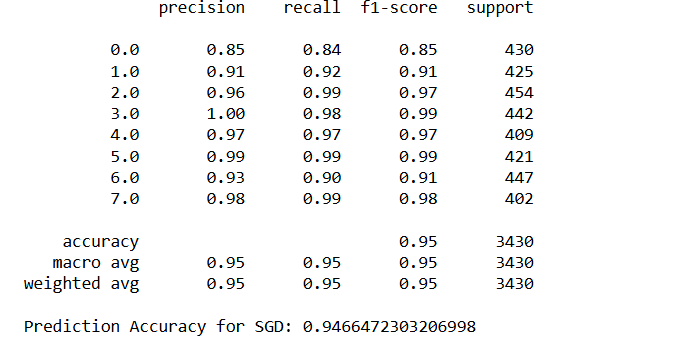
By analysing this confusion matrix, this study observe that the diagonal elements have higher values, indicating a large number of correct predictions. The off-diagonal cells represent incorrect predictions. For instance, cell (1, 2) has a value of 19, meaning 19 instances of class 1 were misclassified as class 2, and cell (4, 1) has a value of 8, indicating 8 instances of class 4 were misclassified as class 1.

These high diagonal values indicate a strong ability to correctly classify instances of each class, though there is room for improvement in reducing misclassifications

**6.3 Stochastic Gradient Descent (SGD):**

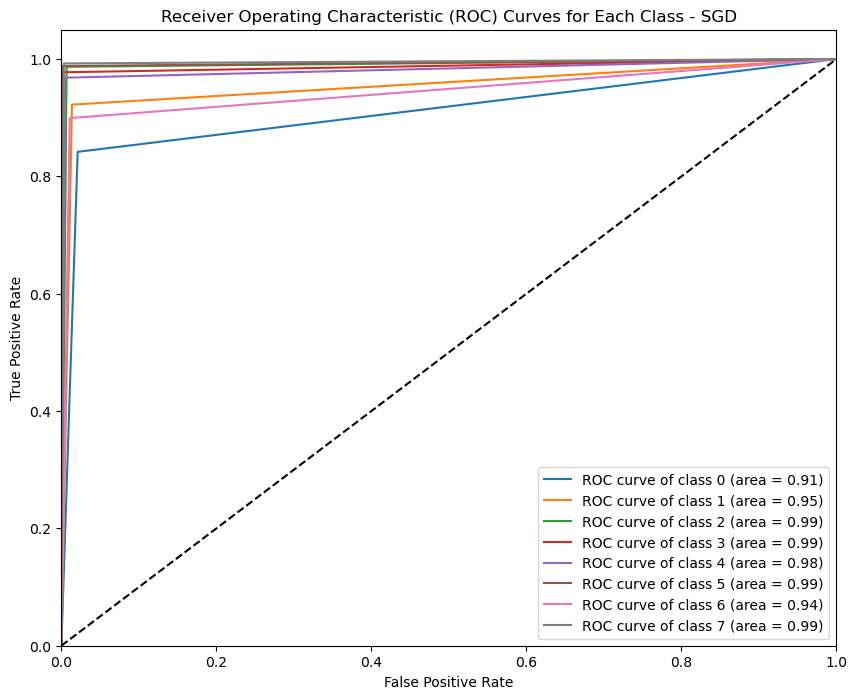
Stochastic Gradient Descent (SGD) serves as an optimization method aimed at minimizing an objective function (Shalev-Shwartz & Ben-David, 2014), such as a loss function used in machine learning models. This technique is particularly beneficial for handling large datasets. Stochastic Gradient Descent (SGD) optimizes model parameters by updating them frequently with a single training example or small batch, rather than the entire dataset. This accelerates convergence, especially for large datasets, and helps escape local minima. For multiclass classification, SGD is advantageous due to its scalability, ability to handle incremental learning, compatibility with multiclass loss functions like cross-entropy, and ease of integrating regularization techniques. This makes SGD efficient and effective for diverse and large-scale classification tasks.

**Table\_5: Classification report for Stochastic Gradient Descent (SGD):**



Bagging with Stochastic Gradient Descent (SGD) shows a training accuracy of 94.71% and a testing accuracy of 94.66%. No over fit found in SGD model. The model underperforms in terms of precision, recall, and F1 scores, which vary significantly across classes. The overall accuracy of the model is 0.946. Notably, the scores for classes 0.0 and 6.0 are particularly low, suggesting that the model struggles to accurately classify these specific classes. This variability in performance indicates that, while the model maintains overall accuracy, it lacks consistency in classifying all categories effectively. Given these issues, it is advisable to explore other models to enhance performance.

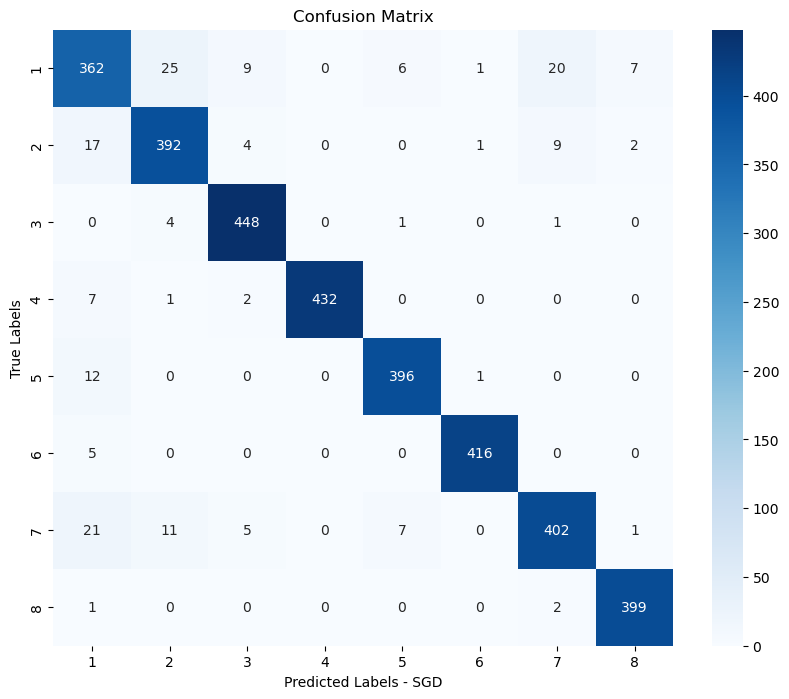
**Figure 10: ROC curve of SGD**



While validating the performance of an SGD classifier, found that the AUC for classes 2, 3, 4, 5, and 7 exceeds 0.98, indicating near-perfect classification for these categories. For classes 0, 1, and 6, the AUCs are slightly lower at 0.91, 0.95, and 0.94, respectively, reflecting slightly weaker but still robust performance. These results suggest that the SGD model effectively distinguishes most classes, achieving overall good performance.

The classifier maintains a AUC above 0.90 across all classes, highlighting its reliability and effectiveness in handling multiclass classification tasks. Despite minor variations in performance between classes, the high AUC values demonstrate the model's capability to provide accurate predictions and robust classification. But there is still room for the improvement.

**Figure 11: Confusion Matrix Stochastic Gradient Descent (SGD)**

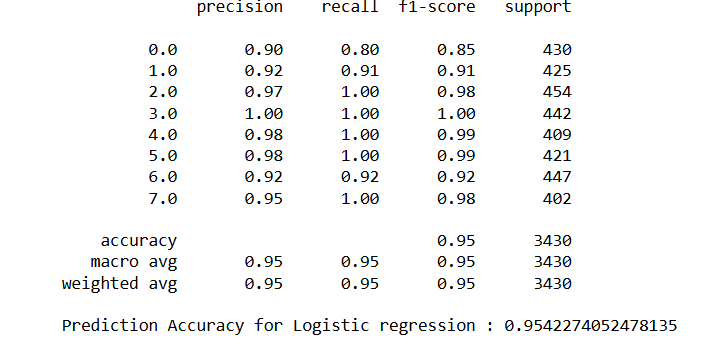


The off-diagonal cells in the confusion matrix represent incorrect predictions. For example, cell (1, 2) has a value of 25, meaning 25 instances of class 1 were misclassified as class 2. Similarly, cell (7, 1) has a value of 21, indicating 21 instances of class 7 were misclassified as class 1, and cell (2, 7) shows 11 instances of class 2 misclassified as class 7. Despite these misclassifications across various classes, the higher values in the diagonal cells indicate a large number of correct predictions. This suggests that the model performs moderately well overall, correctly identifying the majority of instances in each class. While the presence of misclassifications highlights areas for improvement, the strong performance on the diagonal underscores the model's general effectiveness.

**6.4 Logistic Regression:**

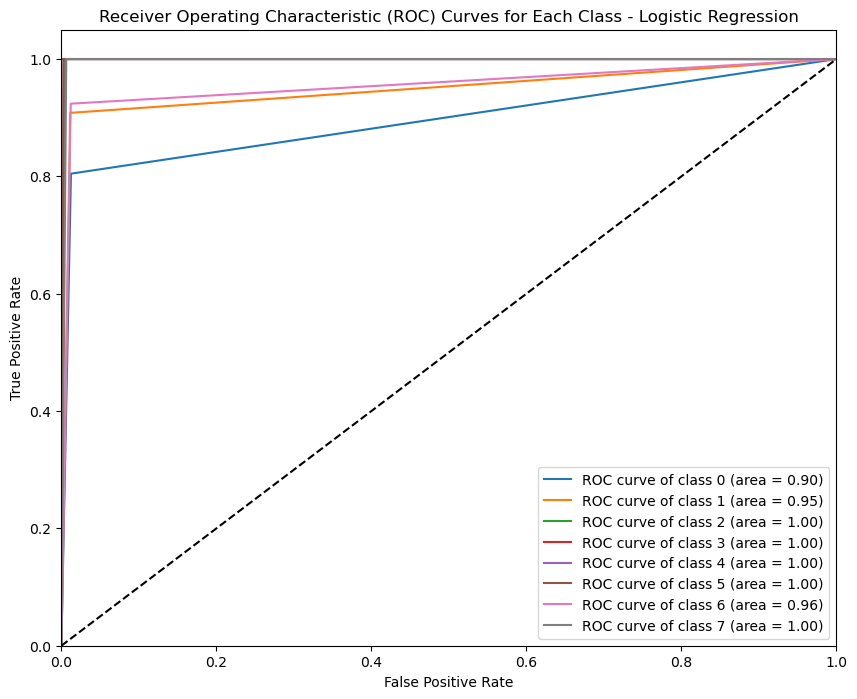
Logistic regression is commonly used to carry out binary classification tasks, but it can be adjusted for multiclass classification using methods like One-vs-Rest (OvR) and Softmax Regression (also known as Multinomial Logistic Regression). In our project employed the OvR method, which is especially advantageous for handling imbalanced datasets. This technique entails creating a binary classifier for each class to determine if a data point belongs to that class. Upon prediction, each classifier assigns a score to the data point, and the class with the highest score is chosen as the final prediction**.**

**Table\_6: Classification report of Logistic regression:**

****

Bagging with Logistic Regression (LR) shows a training accuracy of 96.10% and a testing accuracy of 95.42%. Overall accuracy is 0.954. The classification report reveals strong overall performance, with high precision, recall, and F1 scores across most classes. However, class 0.0 has a lower recall of 0.80, suggesting some difficulty in identifying all instances of this class. Despite this, other classes exhibit consistently high metrics, demonstrating the model's effectiveness and robustness. The minor gap between training and testing accuracies underscores the model's generalization capability. While the overall performance is commendable, improving recall for class 0.0 could further enhance the model's reliability and effectiveness in diverse scenarios.

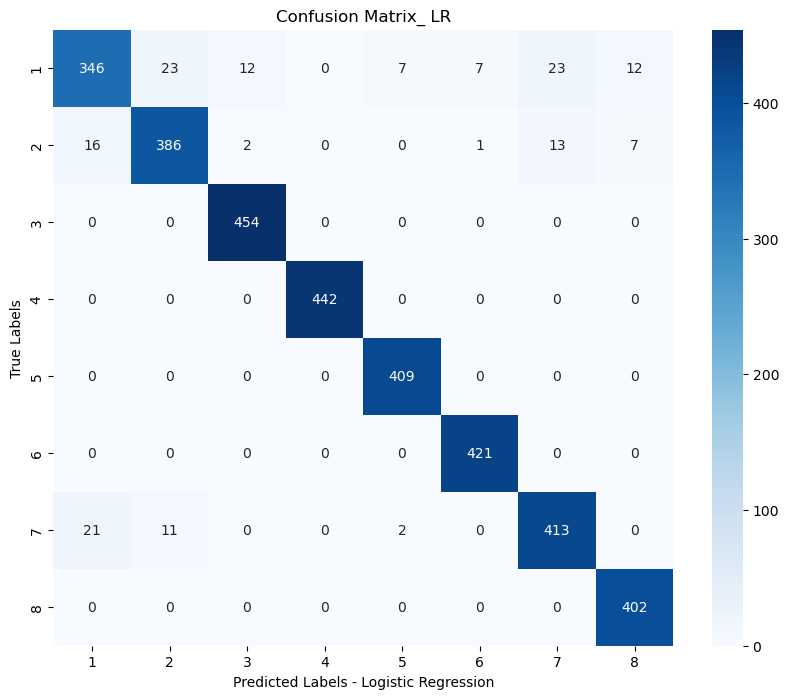
**Figure 12: ROC curve of logistic regression**



Classes 2, 3, 4, 5, and 7 achieved an AUC of 1, indicating perfect classification by the model for these categories. Class 6 has an AUC of 0.96, class 0 has an AUC of 0.90 and class 1 has 0.95. Despite these slight differences, all classes maintain an AUC above 0.9, demonstrating a robust ability to distinguish between positive and negative classes overall. This high level of performance indicates that the model is highly effective in classifying the data accurately across the majority of the classes.

The high AUC scores for most classes highlight the model's proficiency in handling diverse classifications and ensuring reliable predictions. The minor discrepancies in AUC values suggest that while the model is generally very strong, there may be room for fine-tuning to elevate the performance of specific classes. Overall, the consistently high AUC values affirm the model's effectiveness and reliability in accurately classifying the data.

**Figure 13: Confusion Matrix -Logistic Regression**



By analysing the confusion matrix, it is clear that this model predicts almost all classes accurately, with the exception of class 0. The high values aligned diagonally indicate true positive values, while off-diagonal values are minimal compared to previous models. There are only a few true negative values in the matrix. For instance, cell (1, 2) shows 23 false negative values, where instances in class 0 were misclassified as class 1. Additionally, there are instances in class 0 misclassified into other classes except class 4.

Overall, the model shows high efficiency in differentiating between positive and negative values despite these misclassifications. The model's robustness and dependability are demonstrated by the constantly high diagonal values, which show that the majority of examples for each class are properly identified. The overall performance across all classes indicates that the model performs remarkably well, efficiently managing to categorise the data with high accuracy, even while the presence of some false negatives highlights opportunities for development.

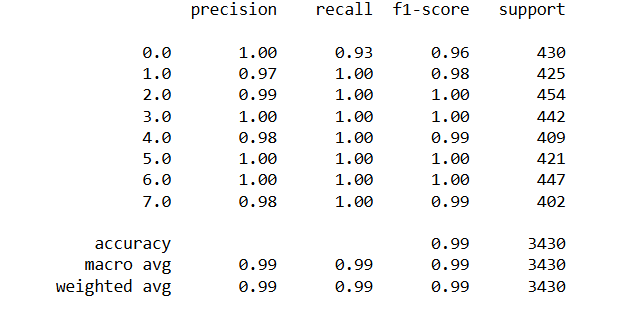
**6.5 Random Forest:**

The random forest stands as a potent algorithm in machine learning, belonging to the realm of ensemble learning. It employs multiple decision trees to enhance prediction accuracy and prevent overfitting. One of its primary techniques, known as bootstrap aggregation or bagging, involves generating multiple datasets by sampling with replacement from the original data, followed by training a decision tree on each dataset. While constructing each decision tree in the random forest, a random subset of features is selected at each node to determine the best split, thereby increasing the diversity of the trees. This additional randomness at each node diminishes the correlation between individual trees, resulting in a more resilient and precise model. For tasks involving classification, the random forest algorithm consolidates the individual predictions of each tree and presents the mode of these predictions as the final result. This method has been verified to yield reliable and consistent outcomes across various applications and random forests are adept at efficiently handling large datasets.

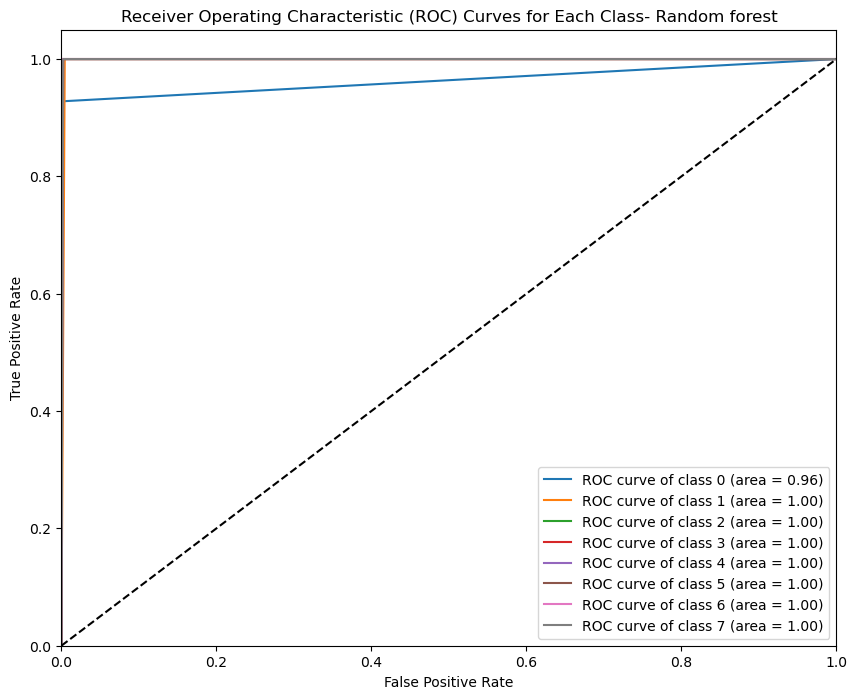
Training Accuracy (99.88%): This shows that the model does remarkably well on the training set of data. It's important to keep in mind, though, that this metric may be excessively optimistic. This measure, Testing Accuracy (99.10%), is more representative of the model's performance in actual use. Even though it's still high, there may be some overfitting shown by the tiny discrepancy between training and testing accuracy.

Metrics like precision, recall, and F1-score offer a more detailed picture of the model's effectiveness across several classes. overall accuracy of the model is 0.990 High scores (above 0.90) in every class indicate strong performance in detecting real positives and reducing false positives and negatives.

**Table 7: Classification report for Random Forest**

****

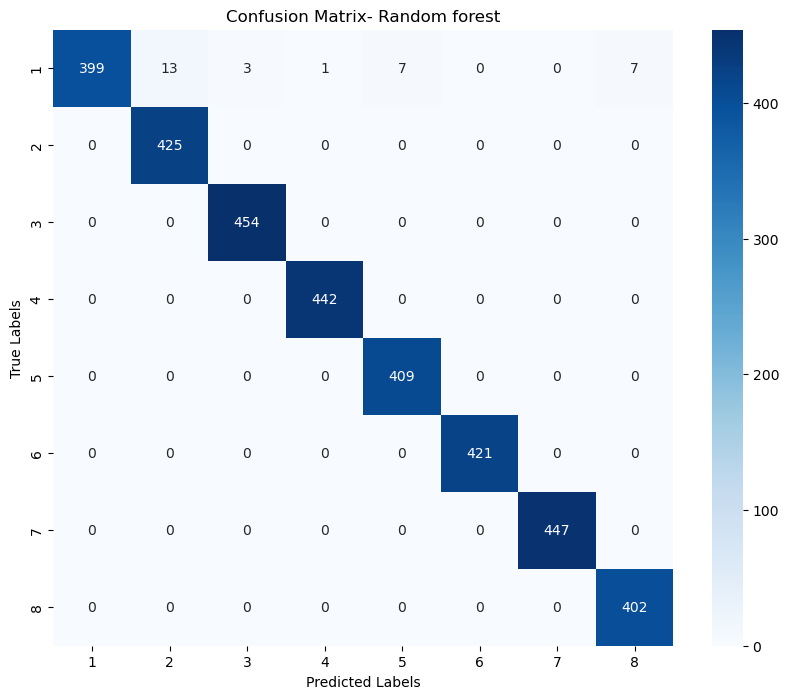
**Figure 14 : Roc curve of Random Forest**



Upon verifying the Random Forest's performance found that the AUC for classes 1 through 7 is extremely high, signifying flawless classification with an AUC of 1. Class 0 shows slightly lower but still strong performance, with an AUC of 0.96. These findings imply that the Random Forest model performs well overall by successfully differentiating the majority of classes. The classifier consistently maintains an impressive AUC above 0.96 across all classes, demonstrating its reliability and effectiveness in handling multiclass classification tasks.

The high AUC values indicate that the Random Forest model is highly adept at distinguishing between positive and negative instances for nearly all classes. This consistency suggests that the model is robust and capable of providing accurate predictions across a diverse set of categories. While there is a minor gap in performance for class 0, the overall results affirm the model's strong classification capabilities, making it a dependable tool for predictive analysis.

**Figure 15: Confusion matrix – Random Forest**



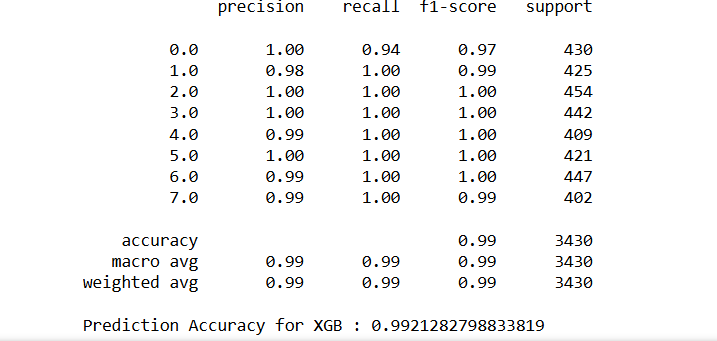
By analysing the confusion matrix, it is clear that this model predicts almost all classes accurately, with the exception of class 0. High values are aligned diagonally, indicating true positive values, while off-diagonal values are minimal compared to previous models. Notably, all true negative values are 0, demonstrating the model's efficiency in distinguishing between positive and negative values. False negatives are found only in class 0, with a total of 27 instances misclassified into other classes. Despite these misclassifications, they are relatively few.

The high diagonal values reflect the model's ability to correctly identify the majority of instances in each class, showcasing its robustness and reliability. The minimal off-diagonal values compared to previous models highlight the improved accuracy and reduced error rates. Overall, this performance indicates that the model works exceptionally well across all classes, providing accurate and reliable classifications. Addressing the misclassifications in class 0 could further enhance the model's already strong performance.

**6.6 XGBoost classifier:**

The XGBoost classifier is well-regarded for its highly efficient implementation of the gradient boosting algorithm, as detailed by Chen and Guestrin in 2016. Extreme Gradient Boosting, or XGBoost, is a powerful machine learning technique that excels in both regression and classification applications. It is a member of the boosting algorithms family that builds a sequence of weak learners (typically decision trees) by repairing the mistakes of its predecessors. Boosting builds a powerful predictor by successively combining models, in contrast to bagging techniques like Random Forest. L1 (Lasso) and L2 (Ridge) regularisation are incorporated into XGBoost to reduce overfitting and improve its robustness against noisy data. It reduces model complexity and boosts performance by removing ineffective branches via sophisticated tree pruning. Furthermore, by figuring out the best split direction, XGBoost can handle missing data and maintain data integrity. Our dataset is highly imbalanced. As per Le et al., 2022 XGBoost classifier is one of the best classifiers for handling data imbalance. Because of its capabilities for distributed and parallel computing, it processes data much more quickly, which makes it extremely effective for handling massive datasets.

**Table\_8 Classification report XGBoost classifier**

****

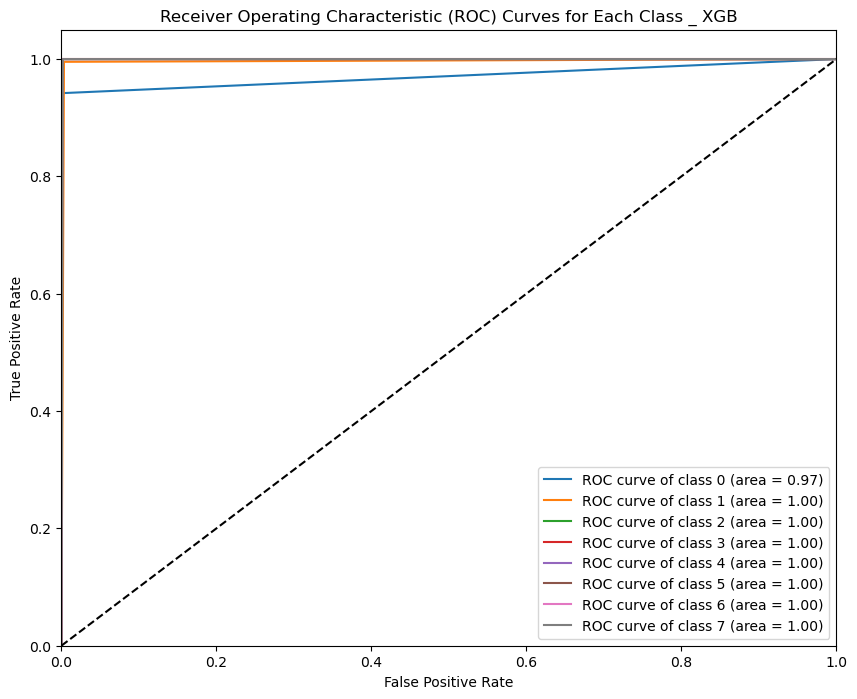
The XGBoost classifier has achieved a training accuracy of 99.98% and a testing accuracy of 99.21%, indicating no overfitting in the data. This demonstrates that the model generalizes well to unseen data.

Precision: Measures the proportion of true positive predictions among all positive predictions. High precision across all classes means the model is highly accurate when it predicts a class. An overall precision of 0.99 means that the model’s positive predictions are highly reliable

Recall (Sensitivity): Measures the proportion of true positive predictions among all actual positives. High recall indicates the model captures nearly all instances of each class. An overall recall of 0.99 indicates that the model successfully identifies almost all actual positive cases. This is particularly important in fields like medical diagnosis, where missing a positive case can have severe consequences.

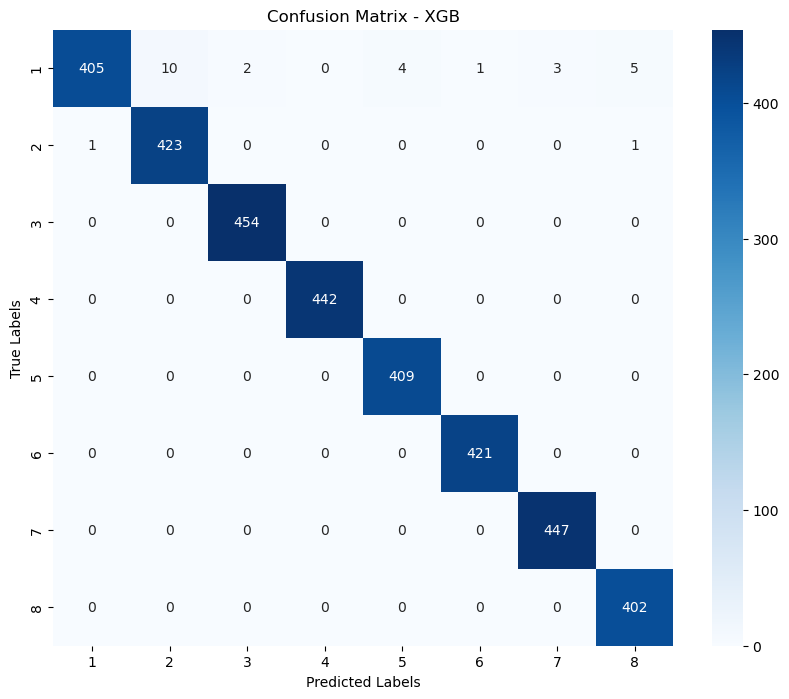
F1 Score: The harmonic means of precision and recall, providing a balance between the two. A high F1 score for all classes signifies a well-rounded performance. The overall F1 score of 0.99 shows a perfect balance between precision and recall, ensuring that the model is both accurate and sensitive. This balance is essential for maintaining high performance across all evaluation metrics. Overall prediction accuracy of this model is 0.992. The XGBoost classifier demonstrates outstanding performance with high accuracy, precision, recall, and F1 scores. The minimal overfitting and high metric values across all classes confirm that this model is robust and reliable for prediction tasks.

**Figure 16– ROC curve of XGBoost classifier**



The analysis of the XGBoost ROC curve reveals that classes 1 through 7 each achieve a perfect AUC of 1.0, signifying flawless classification for these categories. Although class 0 has a slightly lower AUC of 0.97, it still demonstrates a high level of performance. These results indicate that the XGBoost model excels overall, effectively distinguishing between the majority of classes with high accuracy. The consistent AUC of 0.97 across all classes underscores the model's reliability and robust performance. This high AUC value reflects the model’s strong ability to discriminate between positive and negative instances, confirming its effectiveness in handling multiclass classification tasks. Despite the minor reduction in performance for class 0, the model's overall impressive metrics highlight its capability to deliver accurate and dependable predictions across a diverse range of classes.

**Figure 17: Confusion Matrix- XGBoost classifier**



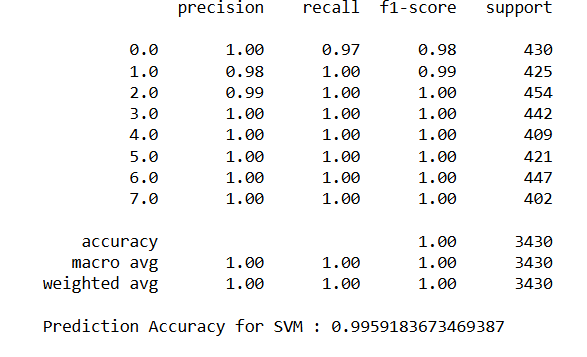
Except for class 0, this model predicts almost all classes accurately, as shown by the analysis of the confusion matrix. Compared to earlier models, off-diagonal values indicating misclassifications are negligible, while the diagonal cells, representing true positives, exhibit substantial values. The model’s ability to effectively distinguish between positive and negative cases is further evidenced by the fact that all true negative values are zero. There are 25 false negatives, where instances from class 0 were misclassified into other classes. Despite this issue with class 0, the model performs remarkably well across the majority of classes. The strong diagonal values highlight its effectiveness in identifying most instances correctly, demonstrating a high level of accuracy and reliability overall. Addressing the misclassification of class 0 could further enhance the model's performance, but the current results underscore its robust and efficient classification capabilities.

**6.7 Support Vector Machine (SVM):**

Support Vector Machine (SVM) is a strong supervised machine learning technique that is widely utilised in classification and outlier detection tasks. It excels in high-dimensional environments and is renowned for its robustness and precision (Janardhanan et al., 2015). The fundamental goal of SVM is to discover the best hyperplane that can successfully split data into distinct groups. In an n-dimensional space, a hyperplane is a (n-1) dimensional subspace that divides the space into two halves. The data points nearest to the hyperplane are referred to as support vectors, and they are crucial in determining the hyperplane's position and orientation. The margin, or distance between the hyperplane and the nearest support vector from either class, is a crucial component in SVM. The approach aims to maximise this margin by ensuring that the hyperplane is correctly positioned to identify the classes.

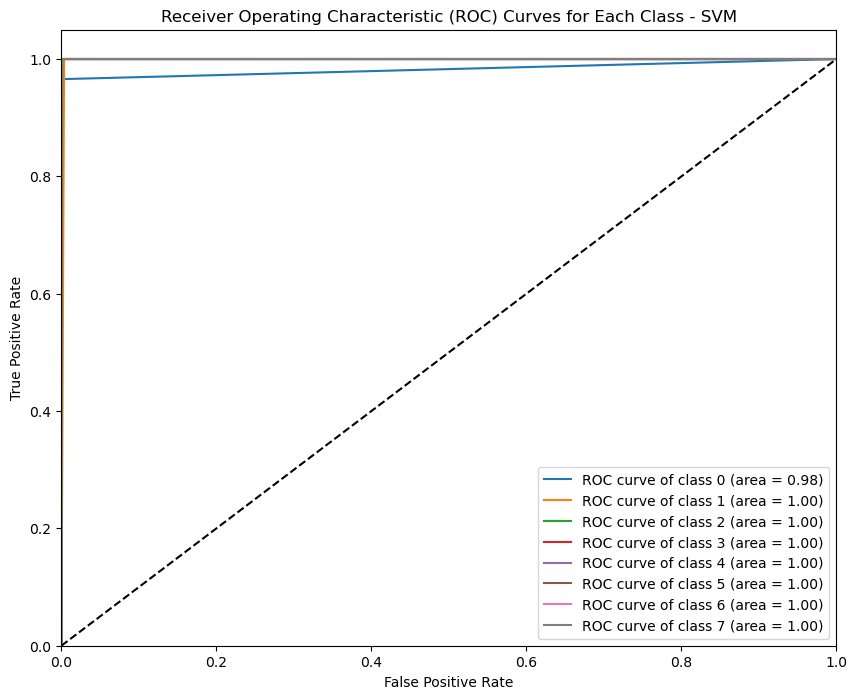
In cases where the data is not linearly separable in the original feature space, SVM utilizes kernel functions to transform the data into a higher-dimensional space, where a hyperplane can effectively separate the classes. Some common kernel functions used in SVM include linear, polynomial, radial basis function (RBF), and sigmoid. By employing these kernel functions, SVM can handle non-linear classification problems and achieve accurate and efficient results.

**Table\_9: Classification report Support Vector Mchine (SVM)**



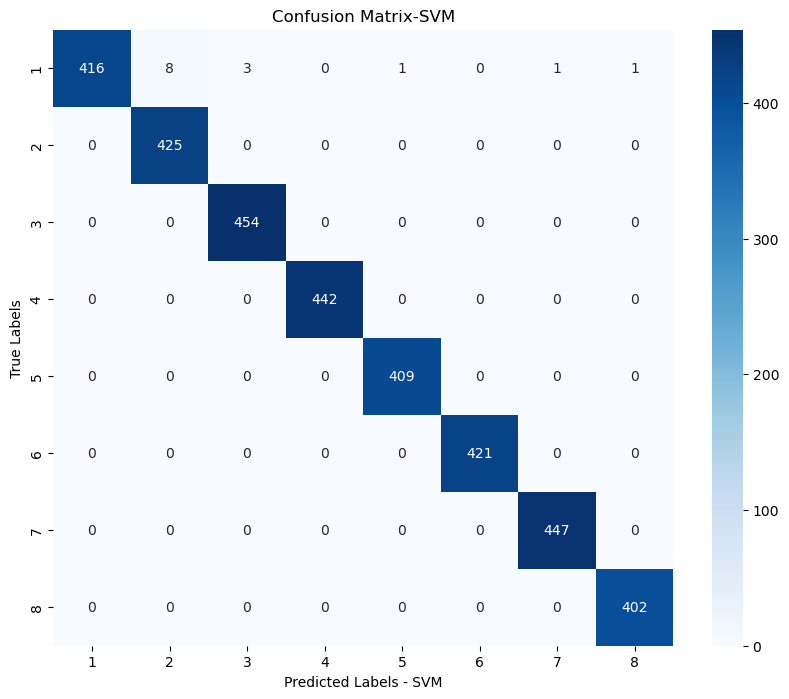
By applying the SVM model, the testing accuracy achieved is 99.59%, while the training accuracy is 99.95%, indicating that the model generalizes exceptionally well with no issues related to overfitting or outliers. The precision, recall, and F1 score for all individual classes are above 0.97, and the overall metrics reach 1.0, showcasing the model's high effectiveness and reliability. The predictive accuracy of the SVM model is 0.995, reflecting its ability to deliver accurate and robust predictions. This impressive performance suggests that the SVM model stands out as one of the best options for classification tasks, providing exceptional accuracy and consistency. Its ability to maintain high performance across both training and testing datasets, without significant overfitting or misclassification issues, underscores its suitability for complex classification problems, making it a top choice for achieving reliable and precise results.

**Figure 18 – ROC curve of Support Vector Mchine (SVM)**



The SVM ROC curve analysis shows perfect classification for classes 1 through 7, with each achieving an AUC of 1.0, indicating flawless performance. Class 0 has a slightly lower AUC of 0.98, but this still reflects a very high level of performance. The high AUC values for most classes demonstrate that the SVM model excels at accurately and consistently differentiating between the majority of classes. The near-perfect AUC for class 0 further highlights the model's remarkable overall performance. These results underscore the model's effectiveness in handling diverse classification tasks, offering exceptional reliability and accuracy across nearly all classes. The SVM model’s ability to achieve such high AUC values showcases its robustness and proficiency in providing accurate predictions, confirming its status as a top-performing classification tool.

**Figure 19: Confusion matrix- Support Vector Mchine (SVM)**



Class 0 is among the nearly all classes that this model correctly predicts, according to the confusion matrix analysis. In contrast to earlier models, the diagonal cells show large values, signifying genuine positives, while the off-diagonal values, which represent inaccurate predictions, are low. This implies that the model is successful in accurately detecting positive examples. The accuracy with which the model can differentiate between positive and negative cases is demonstrated by the fact that all true negative values are zero. Merely 8 cases six from class 2, 3 from class 3, 1 from class 5,1 from class 7 and 1 from class 8 are falsely classed as false negatives inside class 0. There aren't any other incorrect classes. Overall, the model performs exceptionally well across most classes, demonstrating strong accuracy and reliability. This performance confirms the model's effectiveness, making it highly reliable for classification tasks and providing accurate predictions with minimal errors.

**7. Model Comparison:**

**7.1 Accuracy, precision, recall, F1 score comparison**

**Table 10-Model comparison**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 score** |
| Random forest | 0.990 | 0.99 | 0.99 | 0.99 |
| XGBoost classifier: | 0.992 | 0.99 | 0.99 | 0.99 |
| Support Vector Machine (SVM): | 0.995 | 1 | 1 | 1 |
| Stochastic Gradient Descent (SGD): | 0.946 | 0.95 | 0.95 | 0.95 |
| K-Nearest Neighbours (KNN): | 0.962 | 0.96 | 0.96 | 0.96 |
| Logistic Regression | 0.954 | 0.95 | 0.95 | 0.95 |
| Decision Tree | 0.968 | 0.97 | 0.97 | 0.97 |

The following text provides a comparison of the performance of six different classification models: Random Forest, XGBoost, Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), K-Nearest Neighbours (KNN), and Logistic Regression. The comparison is based on four key metrics: Accuracy, Precision, Recall, and F1 score. Among the models, the Support Vector Machine (SVM) stands out with the highest accuracy of 0.995 and perfect scores of 1 for precision, recall, and F1 score, indicating exceptional classification performance. XGBoost follows closely with an accuracy of 0.992 and high precision, recall, and F1 score, making it very effective, particularly in terms of accuracy.

Random Forest also performs exceptionally well with an accuracy of 0.990 and consistent metrics at 0.99 for precision, recall, and F1 score. K-Nearest Neighbours (KNN) demonstrates robust performance with an accuracy of 0.962 and scores of 0.96 for precision, recall, and F1 score, albeit slightly lower than the top models.

Logistic Regression offers solid metrics with an accuracy of 0.954 and precision, recall, and F1 scores at 0.95, serving as a strong baseline. However, Stochastic Gradient Descent (SGD) lags behind with an accuracy of 0.946 and lower metrics at 0.95, indicating relatively weaker performance compared to the other models. From the table, it is clear that SVM and XGBoost show superior performance among the other models.

**7.2 Comparison of AUC of ROC**

When comparing ROC curves of all models, observe that most models have high AUC values across nearly all classes. Specifically, Random Forest, XGBoost (XGB), and Support Vector Machine (SVM) perform exceptionally well, achieving an AUC of 1 for all classes except class 0. For class 0, XGB has an AUC of 0.97, and SVM has an AUC of 0.98. This indicates that while these models perform nearly perfectly, SVM slightly outperforms the others in overall classification accuracy.

In contrast, KNN, Decision Tree, Logistic Regression, and Stochastic Gradient Descent (SGD) classifiers show marginally high AUC values but leave room for improvement, particularly in classes 0 and 1. These models demonstrate respectable performance but fall short of the exceptional results seen with Random Forest, XGB, and SVM.

The detailed analysis of the ROC curves highlights that while all models are effective, SVM stands out as the most reliable, consistently delivering high AUC values and superior discriminatory power across classes. This makes SVM the preferred model for tasks requiring the highest accuracy and performance. Meanwhile, the other models, though still effective, may require further optimization to match the high standards set by SVM, especially for challenging classes like class 0.

**7.3 Confusion matrix analysis**

In conclusion, the confusion matrix analysis reveals that the SVM, XGBoost, and Random Forest models exhibit exceptional performance, achieving near-perfect accuracy across most classes and flawless classification for classes 0 through 7. These models stand out for their robustness and reliability. In contrast, KNN and Decision Tree show moderate performance with notable misclassifications, particularly affecting classes 1 and 0. The SGD model, while effective, has higher misclassification rates compared to the leading models. Overall, SVM, XGBoost, and Random Forest are the top performers, with SVM demonstrating the highest performance among them. This makes SVM the most effective model for accurate and reliable predictions, outperforming the others evaluated in terms of overall effectiveness and consistency.

**8. Result Conclusion:**

To sum up, there are significant variations in the performance of the six classification models Random Forest, XGBoost, SVM, SGD, KNN, and Logistic Regression when compared to one another. The Support Vector Machine (SVM) performs exceptionally well overall, achieving a maximum accuracy of 0.995 and perfect scores of 1 for recall, precision, and F1 score. With a 0.992 accuracy and excellent performance metrics specifically, accuracy XGBoost trails closely behind. With an accuracy of 0.990 and steady precision, recall, and F1 scores of 0.99, Random Forest likewise exhibits excellent performance. With accuracies of 0.963 and 0.962, respectively, Logistic Regression and K-Nearest Neighbours (KNN) and offer reliable but marginally worse performance in comparison to the best models. Among the models, Stochastic Gradient Descent (SGD) performs the worst, with an accuracy of 0.946 and lower scores for precision, recall, and F1.

Random Forest, XGBoost, and SVM all obtain nearly perfect AUC values, according to ROC curve analysis, with SVM doing marginally better than the others, especially for class 0. There is potential for improvement in the decent but less reliable performance of KNN, Decision Tree, and Logistic Regression. SVM, XGBoost, and Random Forest are the three most notable models overall. SVM performs the best, making it the most dependable option for applications needing the highest level of accuracy and classification power.

While the SVM exhibits exceptional performance with high accuracy and perfect recall, precision, and F1 scores in this study, its impressive results may not generalize to other datasets. The performance is closely tied to the specific dataset from the Krasnoyarsk Interdistrict Clinical Hospital, which may possess unique biases or characteristics. If other datasets have different distributions or feature characteristics, the SVM's effectiveness could vary. Additionally, SVM’s performance can be sensitive to hyperparameters and data specifics, potentially impacting its reliability across diverse datasets. To better understand the model's robustness and generalizability, it is crucial to conduct a more comprehensive evaluation using external datasets and additional metrics beyond accuracy, precision, recall, and F1 score. This broader analysis would provide a more accurate assessment of how well the SVM and other models perform in varied real-world scenarios.

**9. Ethical Implications of Results:**

Achieving 100% accuracy in predicting myocardial infarction complications would revolutionize healthcare by enabling early, precise interventions, improving patient outcomes, and optimizing resource allocation, potentially reducing costs. However, such accuracy is unlikely due to the complexities of biological variability and data limitations. In this project, the best-performing model showed an accuracy of 0.995, which is very close to 100%. Incorrect predictions pose significant risks, including false positives that could lead to unnecessary treatments and harm, and false negatives that might result in missed diagnoses and severe complications. Additionally, overfitting, model biases, and lack of transparency in predictions can undermine trust and fairness. Ensuring data privacy, obtaining informed consent, and defining accountability are crucial to managing these risks responsibly. Thus, while the potential for highly accurate predictions is promising, it is essential to address these ethical and practical challenges to ensure beneficial and equitable use in healthcare.

**10. Project Management:**

The planning, assigning, overseeing, and directing of a project to accomplish its goals within the allocated time, financial constraints, and quality standards is all included in project management. It is necessary to maintain the project's direction and guarantee its success. We will go over three important project management topics that are pertinent to this research project in the parts that follow: schedule management, risk management, and quality management.

**10.1 Project Schedule:**

The success of any project hinges on having a well-defined list of work items and a clear project timeline. To achieve this, activities were listed in a work breakdown structure, and timelines were allocated based on preliminary estimates made at the project's inception. Initially, I chose a different topic for the project, but due to the unavailability of the related dataset, I had to change the topic, causing a delay in starting. The report creation was planned to begin concurrently with coding and execution. However, the report work was delayed, and additional time was dedicated to model testing due to issues with overfitting.

**10.2 Risk management:**

Risk management entails detecting, evaluating, and mitigating possible risks to reduce their impact to ensure a project continues on track and meets its goals.

**Table\_11- Risk management**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl No.** | **Identified Risk** | **Impact** | **Risk Severity** | **Mitigation Plan** |
| 1 | Data loss | Could lead to delay in project execution and submission | High | Regular backup on google drive |
| 2 | Challenging to obtain relevant data for the topic | Could delayed the submission | High | Explored various sources to gather the data. |
| 3 | Equipment loss | Execution and submission will be delayed | High | Regular backup of data in external hard disk. |

**10.3 Quality Management**

Quality management employs systematic procedures, continuous monitoring, and improvement to ensure a project meets predefined criteria. At the project's outset, a solid structure is established to keep it on track. Discussions with the supervisor and valuable feedback enhance the reporting structure and overall project quality.

**10.4 Social, Legal, Ethical and Professional Considerations**

Because machine learning programs frequently use datasets containing sensitive information that may be used to identify specific individuals, social, legal, and ethical issues frequently surface. But in this endeavour, there were no moral or legal dilemmas because the dataset was freely available for study and research and was taken from a public website. The Ethical Approval certificate of the project is attached in the appendix.

**11. Discussion:**

During the development of this project, significant challenges were encountered, primarily due to data imbalance. Initially, an under-sampling technique was applied to address this imbalance, but it led to poor performance in minority classes. The model tended to misclassify minority instances as belonging to the majority class, resulting in overfitting on the majority class and underfitting on the minority class. To overcome this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was implemented, which effectively balanced the data.

After balancing the data, overfitting emerged as a major issue during the training phase. To combat this, several techniques were applied, including feature selection using Principal Component Analysis (PCA), 5-fold cross-validation, hyperparameter tuning, and bagging techniques. These strategies were crucial in enhancing the model's performance and are detailed in the methodology section.

In evaluating the six classification models Random Forest, XGBoost, SVM, SGD, KNN, and Logistic Regression there were notable variations in performance. The Support Vector Machine (SVM) excelled with an accuracy of 0.995 and perfect recall, precision, and F1 scores. XGBoost and Random Forest followed closely with accuracies of 0.992 and 0.990, respectively. KNN and Logistic Regression showed slightly lower performance, and SGD performed the worst. ROC curve analysis highlighted SVM, XGBoost, and Random Forest as top models, with SVM marginally better.

However, SVM’s exceptional results may not generalize to other datasets, necessitating further evaluation with external data for robust assessment. These findings underscore the importance of addressing data imbalance and overfitting to develop reliable machine learning models for predicting myocardial infarction complications.

**12. Conclusions:**

**12.1 Achievements:**

Regarding the first objective of this research work, " Which machine learning technique is most accurate in predicting complications arising from myocardial infarction? this study has systematically demonstrated that machine learning is capable of predicting complications from myocardial infarctions, achieving a high accuracy of 0.99. This indicates that machine learning models can perform well on multiclass imbalanced datasets with the help of sampling methods. In this project, the SMOTE technique was used and found to be highly effective for handling heavily imbalanced data.

In addition, the second objective, " Can machine learning effectively handle a multiclass imbalanced dataset related to myocardial infarction?", was also addressed in this research. The findings indicate that the model is capable of handling multiclass imbalanced dataset. By applying special machine learning sampling techniques such as SMOTE can handle the class imbalance effectively. By applying hyper tuning and bagging techniques can overcome the overfitting that caused mainly because of this class imbalance during train test split.

**12.2 Future Work:**

Future work could focus on several areas to enhance model performance. Hyperparameter tuning for each model using Bayesian optimization could yield better results. Feature engineering and selection, based on model-specific importance scores, could improve accuracy and reduce redundancy. Implementing ensemble methods like stacking or blending might combine the strengths of SVM, XGBoost, and Random Forest for superior performance. Data augmentation and synthetic sample generation can address potential imbalances in the dataset. Robust evaluation through k-fold cross-validation and error analysis would ensure model reliability and generalizability. Exploring alternative algorithms or hybrid models, along with testing models in real-world scenarios, could provide new insights and practical performance metrics. Enhancing model interpretability using tools like SHAP or LIME would improve understanding and trust in model predictions.

# **Bibliography and References**

AWS. (2023). *What is SDLC? - Software Development Lifecycle Explained - AWS*. Amazon Web Services, Inc. <https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC>

Belete, D. M., & Huchaiah, M. D. (2021). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, 1–12. <https://doi.org/10.1080/1206212x.2021.1974663>

*Cardiogenic Shock - What is Cardiogenic Shock? | NHLBI, NIH*. (2022, March 24). Www.nhlbi.nih.gov. <https://www.nhlbi.nih.gov/health/cardiogenic-shock#:~:text=Cardiogenic%20shock%2C%20also%20known%20as>

*Cardiovascular diseases (CVDs)*. (n.d.). Www.who.int. Retrieved July 26, 2024, from <https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)?gad_source=1&gclid=CjwKCAjwko21BhAPEiwAwfaQCF4UWt6q28pvMbvVGiTGZGXhCxZM-rMFjwM9R3ICE1ooDfjA6NUn9xoCa8MQAvD_BwE>

Chen, T., & Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, 785–794. <https://doi.org/10.1145/2939672.2939785>

Chen, Z., Shi, J., Pommier, T., Cottin, Y., Salomon, M., Decourselle, T., Lalande, A., & Couturier, R. (2022). Prediction of Myocardial Infarction From Patient Features With Machine Learning. *Frontiers in Cardiovascular Medicine*, *9*. <https://doi.org/10.3389/fcvm.2022.754609>

Chumachenko, D., Butkevych, M., Lode, D., Frohme, M., Schmailzl, K. J. G., & Nechyporenko, A. (2022). Machine Learning Methods in Predicting Patients with Suspected Myocardial Infarction Based on Short-Time HRV Data. *Sensors*, *22*(18), 7033. <https://doi.org/10.3390/s22187033>

Divneet Mandair, Tiwari, P., Simon, S. R., Colborn, K., & Rosenberg, M. (2020). Prediction of incident myocardial infarction using machine learning applied to harmonized electronic health record data. *BMC Medical Informatics and Decision Making*, *20*(1). <https://doi.org/10.1186/s12911-020-01268-x>

Farah, C., Adla, Y. A., & Awad, M. (2022). Can Machine Learning Predict Mortality in Myocardial Infarction Patients within Several Hours of Hospitalization? A Comparative Analysis. *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON)*. <https://doi.org/10.1109/melecon53508.2022.9842984>

Janardhanan, P., L., H., & Sabika, F. (2015). Effectiveness of Support Vector Machines in Medical Data mining. *Journal of Communications Software and Systems*, *11*(1), 25. <https://doi.org/10.24138/jcomss.v11i1.114>

Krueger, R. A. (2014). Focus Groups: A Practical Guide for Applied Research. In *Google Books*. SAGE Publications. <https://books.google.co.uk/books?hl=en&lr=&id=8wASBAAAQBAJ&oi=fnd&pg=PP1&dq=Krueger>

Lam, J. (2003). Enterprise Risk Management: From Incentives to Controls. In *Google Books*. John Wiley & Sons. <https://books.google.co.uk/books?id=wOJ9R7K22hUC&printsec=copyright&redir_esc=y#v=onepage&q&f=false>

Le, T.-T.-H., Oktian, Y. E., & Kim, H. (2022). XGBoost for Imbalanced Multiclass Classification-Based Industrial Internet of Things Intrusion Detection Systems. *Sustainability*, *14*(14), 8707. <https://doi.org/10.3390/su14148707>

Li, F., Xin, H., Zhang, J., Fu, M., Zhou, J., & Lian, Z. (2021). Prediction model of in-hospital mortality in intensive care unit patients with heart failure: machine learning-based, retrospective analysis of the MIMIC-III database. *BMJ Open*, *11*(7), e044779. <https://doi.org/10.1136/bmjopen-2020-044779>

Mechanic, O. J., Gavin, M., & Grossman, S. A. (2023, September 3). *Acute Myocardial Infarction*. Www.ncbi.nlm.nih.gov; StatPearls Publishing. <https://www.ncbi.nlm.nih.gov/books/NBK459269/#:~:text=Acute%20myocardial%20infarction%20(AMI)%20is>

Newaz, A., Mohosheu, M. S., & Al Noman, Md. A. (2023). Predicting complications of myocardial infarction within several hours of hospitalization using data mining techniques. *Informatics in Medicine Unlocked*, *42*, 101361. <https://doi.org/10.1016/j.imu.2023.101361>

Oliveira, M., Seringa, J., Fausto José Pinto, Henriques, R., & Magalhães, T. (2023). Machine learning prediction of mortality in Acute Myocardial Infarction. *BMC Medical Informatics and Decision Making*, *23*(1). <https://doi.org/10.1186/s12911-023-02168-6>

Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine Learning in Medicine. *New England Journal of Medicine*, *380*(14), 1347–1358. <https://doi.org/10.1056/nejmra1814259>

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. <https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf>

UCI Machine Learning Repository. (n.d.). Archive.ics.uci.edu. https://archive.ics.uci.edu/dataset/579/myocardial+infarction+complications

**Appendix A:**

**Dataset over view:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NO** | **Field** | **Description** | **Type** | **Unit** |
| 1 | ID | Record ID (ID): Unique identifier | Integer | ─ |
| 2 | AGE | Age of patient | Integer | ─ |
| 3 | Sex | 0: female, 1: male | Binary |  |
| 4 | INF\_ANAM | Quantity of myocardial infarctions in the anamnesis. 0: zero 1: one 2: two 3: three and more | Categorical |  |
| 5 | STENOK\_AN | ant\_im | Categorical |  |
| 6 | FK\_STENOK | Functional class (FC) of angina pectoris in the last year. 0: there is no angina pectoris 1: I FC 2: II FC 3: III FC 4: IV FC | Categorical |  |
| 7 | IBS\_POST | Coronary heart disease (CHD) in recent weeks, days before admission to hospital 0: none 1: exertional angina pectoris 2: unstable angina pectoris | Categorical |  |
| 8 | IBS\_NASL | Heredity on CHD 0: isn't burdened 1: burdened | Binary |  |
| 9 | GB | Presence of an essential hypertension 0: there is no essential hypertension 1: Stage 1 2: Stage 2 3: Stage 3 | Categorical |  |
| 10 | SIM\_GIPERT | Symptomatic hypertension 0:yes, 1:no | Binary |  |
| 11 | DLIT\_AG | there was no arterial hypertension 1: one year 2: two years 3: three years 4: four years 5: five years 6: 6-10 years 7: more than 10 years | Categorical |  |
| 12 | ZSN\_A | Presence of chronic Heart failure (HF) in the anamnesis: Partially ordered attribute: there are two lines of severities: 0<1<2<4, 0<1<3<4. State 4 means simultaneous states 2 and 3 0: there is no chronic heart failure 1: I stage 2: II stage (heart failure due to right ventricular systolic dysfunction) 3: II stage (heart failure due to left ventricular systolic dysfunction) 4: IIB stage (heart failure due to left and right ventricular systolic dysfunction) | Categorical |  |
| 13 | nr\_11 | Observing of arrhythmia in the anamnesis  0 no, 1 yes | Binary |  |
| 14 | nr\_01 | Premature atrial contractions in the anamnesis  0 no, 1 yes | Binary |  |
| 15 | nr\_02 | Premature ventricular contractions in the anamnesis 0 no, 1 yes | Binary |  |
| 16 | nr\_03 | Paroxysms of atrial fibrillation in the anamnesis  0 no, 1 yes | Binary |  |
| 17 | nr\_04 | A persistent form of atrial fibrillation in the anamnesis 0 no, 1 yes | Binary |  |
| 18 | nr\_07 | Ventricular fibrillation in the anamnesis 0 no, 1 yes | Binary |  |
| 19 | nr\_08 | Ventricular paroxysmal tachycardia in the anamnesis: 0 no, 1 yes | Binary |  |
| 20 | np\_01 | First-degree AV block in the anamnesis  : 0 no, 1 yes | Binary |  |
| 21 | np\_04 | Third-degree AV block in the anamnesis  : 0 no, 1 yes | Binary |  |
| 22 | np\_05 | LBBB (anterior branch) in the anamnesis  : 0 no, 1 yes | Binary |  |
| 23 | np\_07 | Incomplete LBBB in the anamnesis: 0 no, 1 yes | Binary |  |
| 24 | np\_08 | Complete LBBB in the anamnesis Complete LBBB in the anamnesis : 0 no, 1 yes | Binary |  |
| 25 | np\_09 | Incomplete RBBB in the anamnesis: 0 no, 1 yes | Binary |  |
| 26 | np\_10 | Complete RBBB in the anamnesis : 0 no, 1 yes | Binary |  |
| 27 | endocr\_01 | Diabetes mellitus in the anamnesis :0 no, 1 yes | Binary |  |
| 28 | endocr\_02 | Obesity in the anamnesis :0 no, 1 yes | Binary |  |
| 29 | endocr\_03 | Thyrotoxicosis in the anamnesis :0 no, 1 yes | Binary |  |
| 30 | zab\_leg\_01 | Chronic bronchitis in the anamnesis :0 no, 1 yes | Binary |  |
| 31 | zab\_leg\_02 | Obstructive chronic bronchitis in the anamnesis  :0 no, 1 yes | Binary |  |
| 32 | zab\_leg\_03 | Bronchial asthma in the anamnesis :0 no, 1 yes | Binary |  |
| 33 | zab\_leg\_04 | Chronic pneumonia in the anamnesis :0 no, 1 yes | Binary |  |
| 34 | zab\_leg\_06 | Pulmonary tuberculosis in the anamnesis  :0 no, 1 yes | Binary |  |
| 35 | O\_L\_POST | Pulmonary edema at the time of admission to intensive care unit :0 no, 1 yes | Integer |  |
| 36 | S\_AD\_KBRIG | Systolic blood pressure according to Emergency Cardiology Team | Integer | mmHg |
| 37 | D\_AD\_KBRIG | Diastolic blood pressure according to Emergency Cardiology Team | Integer | mmHg |
| 38 | S\_AD\_ORIT | Systolic blood pressure according to intensive care unit | Integer | mmHg |
| 39 | D\_AD\_ORIT | Diastolic blood pressure according to intensive care unit | Integer | mmhg |
| 40 | K\_SH\_POST | Cardiogenic shock at the time of admission to intensive care unit :0 no, 1 yes | Integer |  |
| 41 | MP\_TP\_POST | Paroxysms of atrial fibrillation at the time of admission to intensive care unit, (or at a pre-hospital stage) :0 no, 1 yes | Binary |  |
| 42 | SVT\_POST | Paroxysms of supraventricular tachycardia at the time of admission to intensive care unit, (or at a pre-hospital stage :0 no, 1 yes | Binary |  |
| 43 | GT\_POST | Paroxysms of ventricular tachycardia at the time of admission to intensive care unit, (or at a pre-hospital stage) :0 no, 1 yes | Binary |  |
| 44 | FIB\_G\_POST | Ventricular fibrillation at the time of admission to intensive care unit, (or at a pre-hospital stage) :0 no, 1 yes | Binary |  |
| 45 | ant\_im | Presence of an anterior myocardial infarction (left ventricular) (ECG changes in leads V1: V4 ) 0: there is no infarct in this location 1: QRS has no changes 2: QRS is like QR-complex 3: QRS is like Qr-complex 4: QRS is like QS-complex | Categorical |  |
| 46 | lat\_im | Presence of a lateral myocardial infarction (left ventricular) (ECG changes in leads V5: V6 , I, AVL) 0: there is no infarct in this location 1: QRS has no changes 2: QRS is like QR-complex 3: QRS is like Qr-complex 4: QRS is like QS-complex | Categorical |  |
| 47 | inf\_im | Presence of an inferior myocardial infarction (left ventricular) (ECG changes in leads III, AVF, II). 0: there is no infarct in this location 1: QRS has no changes 2: QRS is like QR-complex 3: QRS is like Qr-complex 4: QRS is like QS-complex | Categorical |  |
| 48 | post\_im | Presence of a posterior myocardial infarction (left ventricular) (ECG changes in V7: V9, reciprocity changes in leads V1 – V3) 0: there is no infarct in this location 1: QRS has no changes 2: QRS is like QR-complex 3: QRS is like Qr-complex 4: QRS is like QS-complex | Categorical |  |
| 49 | IM\_PG\_P | Presence of a right ventricular myocardial infarction :0 no, 1 yes | Binary |  |
| 50 | ritm\_ecg\_p\_01 | ECG rhythm at the time of admission to hospital: sinus (with a heart rate 60-90)  :0 no, 1 yes | Binary |  |
| 51 | ritm\_ecg\_p\_02 | ECG rhythm at the time of admission to hospital: atrial fibrillation :0 no, 1 yes | Binary |  |
| 52 | ritm\_ecg\_p\_04 | ECG rhythm at the time of admission to hospital: atrial :0 no, 1 yes | Binary |  |
| 53 | ritm\_ecg\_p\_06 | ECG rhythm at the time of admission to hospital: idioventricular :0 no, 1 yes | Binary |  |
| 54 | ritm\_ecg\_p\_07 | ECG rhythm at the time of admission to hospital: sinus with a heart rate above 90 (tachycardia) :0 no, 1 yes | Binary |  |
| 55 | ritm\_ecg\_p\_08 | ECG rhythm at the time of admission to hospital: sinus with a heart rate below 60 (bradycardia) :0 no, 1 yes | Binary |  |
| 56 | n\_r\_ecg\_p\_01 | Premature atrial contractions on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 57 | n\_r\_ecg\_p\_02 | Frequent premature atrial contractions on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 58 | n\_r\_ecg\_p\_03 | Premature ventricular contractions on ECG at the 59time of admission to hospital :0 no, 1 yes | Binary |  |
| 59 | n\_r\_ecg\_p\_04 | Frequent premature ventricular contractions on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 60 | n\_r\_ecg\_p\_05 | Paroxysms of atrial fibrillation on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 61 | n\_r\_ecg\_p\_06 | Persistent form of atrial fibrillation on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 62 | n\_r\_ecg\_p\_08 | Paroxysms of supraventricular tachycardia on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 63 | n\_r\_ecg\_p\_09 | Paroxysms of ventricular tachycardia on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 64 | n\_r\_ecg\_p\_10 | Ventricular fibrillation on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 65 | n\_p\_ecg\_p\_01 | Sinoatrial block on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 66 | n\_p\_ecg\_p\_03 | First-degree AV block on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 67 | n\_p\_ecg\_p\_04 | Type 1 Second-degree AV block (Mobitz I/Wenckebach) on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 68 | n\_p\_ecg\_p\_05 | Type 2 Second-degree AV block (Mobitz II/Hay) on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 69 | n\_p\_ecg\_p\_06 | Third-degree AV block on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 70 | n\_p\_ecg\_p\_07 | LBBB (anterior branch) on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 71 | n\_p\_ecg\_p\_08 | LBBB (posterior branch) on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 72 | n\_p\_ecg\_p\_09 | Incomplete LBBB on ECG at the time of admission to hospital:0 no, 1 yes | Binary |  |
| 73 | n\_p\_ecg\_p\_10 | Complete LBBB on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 74 | n\_p\_ecg\_p\_11 | Incomplete RBBB on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 75 | n\_p\_ecg\_p\_12 | Complete RBBB on ECG at the time of admission to hospital :0 no, 1 yes | Binary |  |
| 76 | fibr\_ter\_01 | Fibrinolytic therapy by Сеliasum 750k :0 no, 1 yes | Binary |  |
| 77 | fibr\_ter\_02 | Fibrinolytic therapy by Сеliasum 1m IU :0 no, 1 yes | Binary |  |
| 78 | fibr\_ter\_03 | Fibrinolytic therapy by Сеliasum 3m IU :0 no, 1 yes | Binary |  |
| 79 | fibr\_ter\_05 | Fibrinolytic therapy by Streptase :0 no, 1 yes | Binary |  |
| 80 | fibr\_ter\_06 | Fibrinolytic therapy by Сеliasum 500k :0 no, 1 yes | Binary |  |
| 81 | fibr\_ter\_07 | Fibrinolytic therapy by Сеliasum 250k 500k :0 no, 1 yes | Binary |  |
| 82 | fibr\_ter\_08 | Fibrinolytic therapy by Сеliasum 1.5m IU 500k :0 no, 1 yes | Binary |  |
| 83 | GIPO\_K | Hypokalemia ( < 4 mmol/L) :0 no, 1 yes | Binary |  |
| 84 | K\_BLOOD | Serum potassium content | Continuous | mmol/L |
| 85 | GIPER\_NA | Increase of sodium in serum (more than 150 mmol/L) | Binary |  |
| 86 | NA\_BLOOD | Serum sodium content | Continuous | IU/L |
| 87 | ALT\_BLOOD | Serum AlAT content (ALT\_BLOOD) | Continuous | IU/L |
| 88 | AST\_BLOOD | Serum AsAT content | Continuous | IU/L |
| 89 | KFK\_BLOOD | Serum CPK content | Continuous | IU/L |
| 90 | L\_BLOOD | White blood cell count | Continuous | billions per liter |
| 91 | ROE | ESR (Erythrocyte sedimentation rate) | Continuous | MM |
| 92 | TIME\_B\_S | Time elapsed from the beginning of the attack of CHD to the hospital 1: less than 2 hours 2: 2-4 hours 3: 4-6 hours 4: 6-8 hours 5: 8-12 hours 6: 12-24 hours 7: more than 1 days 8: more than 2 days 9: more than 3 days | Categorical |  |
| 93 | R\_AB\_1\_n | Relapse of the pain in the first hours of the hospital period 0: there is no relapse 1: only one 2: 2 times 3: 3 or more times |  |  |
| 94 | R\_AB\_2\_n | Relapse of the pain in the second day of the hospital period 0: there is no relapse 1: only one 2: 2 times 3: 3 or more times | Categorical |  |
| 95 | R\_AB\_3\_n | Relapse of the pain in the third day of the hospital period 0: there is no relapse 1: only one 2: 2 times 3: 3 or more times | Categorical |  |
| 96 | NA\_KB | Use of opioid drugs by the Emergency Cardiology Team :0 No, 1 Yes | Binary |  |
| 97 | NOT\_NA\_KB | Use of NSAIDs by the Emergency Cardiology Team :0 No, 1 Yes | Binary |  |
| 98 | LID\_KB | Use of lidocaine by the Emergency Cardiology Team :0 No, 1 Yes | Binary |  |
| 99 | NITR\_S | Use of liquid nitrates in the ICU | Integer |  |
| 100 | NA\_R\_1\_n | Use of opioid drugs in the ICU in the first hours of the hospital period | Integer |  |
| 101 | NA\_R\_2\_n | Use of opioid drugs in the ICU in the second day of the hospital period | Integer |  |
| 102 | NA\_R\_3\_n | Use of opioid drugs in the ICU in the third day of the hospital period | Integer |  |
| 103 | NOT\_NA\_1\_n | Use of NSAIDs in the ICU in the first hours of the hospital period 0: no 1: once 2: twice 3: three times 4: four or more times | Categorical |  |
| 104 | NOT\_NA\_2\_n | Use of NSAIDs in the ICU in the second day of the hospital period | Integer |  |
| 105 | NOT\_NA\_3\_n | Use of NSAIDs in the ICU in the third day of the hospital period | Integer |  |
| 106 | LID\_S\_n | Use of lidocaine in the ICU :0 No, 1 Yes | Binary |  |
| 107 | B\_BLOK\_S\_n | Use of beta-blockers in the ICU :0 No, 1 Yes | Binary |  |
| 108 | ANT\_CA\_S\_n | Use of calcium channel blockers in the ICU :0 No, 1 Yes | Binary |  |
| 109 | GEPAR\_S\_n | Use of а anticoagulants (heparin) in the ICU :0 No, 1 Yes | Binary |  |
| 110 | ASP\_S\_n | Use of acetylsalicylic acid in the ICU :0 No, 1 Yes | Binary |  |
| 111 | TIKL\_S\_n | Use of Ticlid in the ICU :0 No, 1 Yes | Binary |  |
| 112 | TRENT\_S\_n | Use of Trental in the ICU :0 No, 1 Yes | Binary |  |
| 113 | FIBR\_PREDS | Atrial fibrillation:0 No, 1 Yes | Binary |  |
| 114 | PREDS \_TAH | Supraventricular tachycardia :0 No, 1 Yes | Binary |  |
| 115 | JELUD\_TAH | Ventricular tachycardia :0 No, 1 Yes | Binary |  |
| 116 | FIBR\_JELUD | Ventricular fibrillation :0 No, 1 Yes | Binary |  |
| 117 | A\_V\_BLOK | Third-degree AV block :0 No, 1 Yes | Binary |  |
| 118 | OTEK\_LANC | Pulmonary edema :0 No, 1 Yes | Binary |  |
| 119 | RAZRIV | Myocardial rupture :0 No, 1 Yes | Binary |  |
| 120 | DRESSLER | Dressler syndrome :0 No, 1 Yes | Binary |  |
| 121 | ZSN | Chronic heart failure :0 No, 1 Yes | Binary |  |
| 122 | REC\_IM | Relapse of the myocardial infarction :0 No, 1 Yes | Binary |  |
| 123 | P\_IM\_STEN | Post-infarction angina :0 No, 1 Yes | Binary |  |
| 124 | LET\_IS | Lethal outcome (cause) 0: unknown (alive) 1: cardiogenic shock 2: pulmonary edema 3: myocardial rupture 4: progress of congestive heart failure 5: thromboembolism 6: asystole 7: ventricular fibrillation | Categorical |  |