# ABSTRACT

Heart disease remains a leading cause of morbidity and mortality globally. Early prediction of heart attack risk is imperative for timely intervention and prevention. This study explores the application of Automated Machine Learning (AutoML) to predict heart attack risk, leveraging a comprehensive dataset of relevant clinical and demographic features. The proposed framework involves data preprocessing, AutoML configuration, and model evaluation. AutoML tools, including Google AutoML and Auto-Sklearn, are employed to automate the model selection and hyperparameter tuning process. Ethical considerations, interpretability, and compliance with healthcare regulations are integral components of the study. The resulting predictive model is assessed for accuracy, interpretability, and clinical relevance. The findings contribute to the ongoing effort to enhance heart disease risk assessment and underscore the importance of ethical and transparent AI practices in healthcare applications.

**Keywords: AUTOML, Heart disease, Optimization, Accuracy, Hyperparameters, Automation.**

# CONTENTS


| S.NO | PAGE NO |
|---|---|
| 1. Introduction | 1 |
| 1.1. Motivation | 1 |
| 1.2. Problem Definition | 1 |
| 1.3. Objective of the Project | 2 |
| 2. Literature Survey | 3 |
| 3. Analysis | 4 |
| 3.1. Existing System | 4 |
| 3.2. Proposed System | 5 |
| 3.3. System Requirement Specification | 6 |
| 3.3.1 Purpose | 6 |
| 3.3.2 Scope | 7 |
| 3.3.3 Overall Description | 7 |
| 4. Implementation | 8 |
| 4.1. List of Files and Description | 9 |
| 4.2. Dataset | 9 |
| 5. Experimental result | 11 |
| 5.1.Experiment setup | 11 |
| 5.2. Parameter with formulas | 13 |
| 5.3. Sample Code | 14 |
| 6. Discussion of result | 24 |
| 6.1.Test cases | 24 |
| 6.2.Screen shots | 25 |
| 7. Conclusion | 29 |
| 8. Future Enhancement | 30 |
| 9. Bibliography | 31 |

List of figures

Fig 4.1.Dataset 10

Fig 5.1.Heart Heart attack rick prediction 12

List of tables

Table 6.1. Testcases 24

# 1. INTRODUCTION

Cardiovascular diseases, including heart attacks, continue to be a major global health concern, imposing a substantial burden on individuals and healthcare systems. Timely identification of individuals at risk of a heart attack is critical for implementing preventive measures and improving patient outcomes. With advancements in machine learning, particularly Automated Machine Learning (AutoML), there is an opportunity to enhance the accuracy and efficiency of heart attack risk prediction.

This project aims to leverage the power of AutoML to develop a robust and automated system for predicting heart attack risk. AutoML offers the advantage of automating the complex process of model selection, hyperparameter tuning, and feature engineering, making it accessible to healthcare professionals and researchers without extensive machine learning expertise.

The dataset used in this study comprises a diverse set of clinical and demographic features, including age, gender, cholesterol levels, blood pressure, and family history. These features are known to play a crucial role in cardiovascular health and are essential for accurate risk assessment.

## 1.1 Motivation

The driving force behind this project lies in the compelling need to transform the landscape of heart attack prediction, a critical facet of cardiovascular health. With cardiovascular diseases persisting as a leading cause of global mortality, there is a clear urgency to develop innovative tools that can enhance early detection and intervention. The motivation stems from the potential of advanced machine learning techniques, particularly Automated Machine Learning (AutoML), to democratize the creation of accurate predictive models. By automating complex processes such as model selection and hyperparameter tuning, we aim to empower healthcare professionals with an accessible and powerful tool for identifying individuals at risk of heart attacks. The project is deeply rooted in the ethos of preventive healthcare, recognizing the significance of timely interventions in mitigating the impact of cardiovascular diseases. Ethical considerations are central to our motivation, guiding the project to prioritize fairness, transparency, and accountability in the deployment of machine learning in the healthcare domain. Ultimately, this project aspires to contribute to global health outcomes by delivering a reliable, interpretable, and ethically sound solution for heart attack risk prediction, thereby improving patient care and outcomes on a broader scale.

## 1.2 Problem Definition

The challenge addressed by this project is the need for accurate and timely prediction of heart attack risk, a critical aspect of cardiovascular health. Cardiovascular diseases, including heart attacks, remain a leading cause of global morbidity and mortality. The existing methods for risk assessment often lack the precision and efficiency needed for early intervention. Consequently, there is a compelling problem to be addressed: the inadequacy of current approaches in providing healthcare professionals with a reliable and automated tool for predicting heart attack risk.

## 1.3 Objective

1. Automated Model Selection: Utilizing AutoML tools to automatically select the most suitable machine learning model for heart attack risk prediction, considering various algorithms and configurations.
2. Feature Importance Analysis: Investigating the contribution of individual features to the predictive model, providing insights into the factors that significantly influence heart attack risk.
3. Ethical Considerations: Integrating ethical considerations into the model development process, addressing potential biases, and ensuring fairness in predictions, especially in the context of healthcare.
4. Compliance and Interpretability: Ensuring that the developed model complies with healthcare regulations, such as data privacy standards (e.g., HIPAA), and prioritizing interpretability for transparency in decision-making.

# 2. LITERATURE SURVEY

The literature survey for this project involves a comprehensive exploration of existing research and studies across various domains relevant to heart attack risk prediction using Automated Machine Learning (AutoML). Firstly, a review of literature on cardiovascular health and risk factors provides insights into the fundamental elements impacting heart attack susceptibility, such as age, gender, cholesterol levels, blood pressure, and family history. Additionally, an examination of traditional risk assessment methods, including established scoring systems like the Framingham Risk Score and the ACC/AHA Risk Calculator, contributes a baseline understanding of existing methodologies.

In the realm of machine learning applications in cardiovascular health, the survey delves into studies that have employed traditional machine learning algorithms to predict heart attack risk based on clinical and demographic data. Furthermore, the investigation extends to literature on the application of AutoML frameworks in healthcare, aiming to identify studies leveraging tools like Google AutoML, Auto-Sklearn, or similar platforms for predictive modeling in medical contexts.

Given the critical nature of healthcare applications, special attention is paid to literature discussing interpretable machine learning models. Understanding how transparency and interpretability can be achieved in predictive models, particularly in the context of heart attack risk prediction, forms a crucial component of the survey. The exploration then extends to studies addressing the challenges of bias and fairness in machine learning models, with a focus on healthcare contexts and strategies for mitigating biases to ensure equitable predictions.

Ethical considerations take center stage in the literature survey, with a specific focus on research discussing privacy, informed consent, and adherence to healthcare regulations, including HIPAA. The survey also encompasses studies highlighting challenges in deploying machine learning models in real-world healthcare settings, considering factors such as data integration, scalability, etc. Finally, the literature survey seeks insights from recent advances and future directions in predictive modeling for cardiovascular health, identifying cutting-edge techniques and proposed approaches to enhance accuracy and usability.

# 3. ANALYSIS

## 3.1 Existing System

existing systems for heart attack risk prediction often leverage traditional risk assessment tools, machine learning models (such as logistic regression and decision trees), deep learning approaches, AutoML platforms like Google AutoML, and ensemble learning techniques to enhance predictive accuracy. Interpretable machine learning models and the incorporation of omics data are also explored for transparency and personalized risk assessment. For the latest advancements, please refer to recent literature and research in the field.

## 3.2 Proposed System

The proposed system envisions a comprehensive and automated approach to heart attack risk prediction, leveraging the power of Automated Machine Learning (AutoML) to enhance accuracy, accessibility, and interpretability. The system's architecture involves several key components:

- **Data Integration and Preprocessing:** The system will integrate a diverse and comprehensive dataset containing essential clinical and demographic features related to cardiovascular health. Robust preprocessing techniques will be employed to handle missing data, normalize numerical features, and encode categorical variables, ensuring the dataset's suitability for training machine learning models.

- **Automated Model Selection with AutoML**: The heart of the proposed system lies in the use of AutoML platforms, such as Google AutoML or equivalent tools. These platforms automate the model selection process, exploring various algorithms, hyperparameters, and preprocessing steps to identify the most effective combination for heart attack risk prediction. This approach not only saves time but also makes advanced machine learning accessible to healthcare professionals with varying levels of expertise.

- **Ethical Considerations and Bias Mitigation:** The proposed system places a strong emphasis on ethical considerations, addressing potential biases in the data and the model. Techniques for mitigating biases will be implemented to ensure fair and equitable predictions. This includes transparency in the decision-making process and healthcare

regulations, promoting a trustworthy and ethical deployment of the predictive model.

- **Interpretability and Explainability:** Recognizing the critical need for interpretability in healthcare applications, the system will prioritize models that offer transparency and explainability. Decision tree-based models, for instance, provide insights into the factors influencing predictions, enabling healthcare professionals to understand and trust the model's outputs.

- **Continuous Monitoring and Model Updates:** The proposed system will incorporate mechanisms for continuous monitoring of model performance over time. This involves evaluating the model's predictions against real-world outcomes and updating the model as needed. This iterative process ensures that the predictive capabilities of the system remain relevant and effective in the dynamic landscape of cardiovascular health.

- **User-Friendly Interface:** To make the system accessible to healthcare professionals, a user-friendly interface will be developed. This interface will allow users to input patient data, receive risk predictions, and interpret the model's insights. The goal is to facilitate seamless integration into existing healthcare workflows.

- In summary, the proposed system is designed to be a sophisticated yet accessible tool for heart attack risk prediction. By integrating cutting-edge AutoML techniques with ethical considerations, interpretability, and continuous monitoring, the system aims to contribute significantly to proactive healthcare strategies and the improvement of patient outcomes in the domain of cardiovascular health.

## 3.3 System Requirement Specification

**Operating system:** Windows

**Language:** Python 3.10

**IDE:** Google colab

**LIBRARIES :** Pandas,Numpy, Seaborn, Matplotlib, EVALML

**PANDAS :** Pandas is a powerful Python library for data manipulation and analysis. It provides data structures like DataFrames and Series, making it easy to work with structured data. Pandas is widely used for tasks such as data cleaning, transformation, and exploration in data science and analysis projects.

**NUMPY :** NumPy is a fundamental library for numerical computing in Python. It offers support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions. NumPy is the backbone of many scientific and data analysis libraries in Python, enabling efficient data manipulation and mathematical operations.

**SEABORN :** Seaborn is a data visualization library built on top of Matplotlib. It provides a high-level interface for creating aesthetically pleasing statistical graphics. Seaborn simplifies the creation of complex visualizations like scatter plots, heatmaps, and distribution plots, making it a popular choice for data visualization in Python.

**MATPLOT :** Matplotlib is a versatile and widely-used library for creating static, animated, or interactive visualizations in Python. It offers fine-grained control over plot customization and is the foundation for many other plotting libraries.

**EVALML :** EVALML is an open-source autoML library written in python that automates a large part of the machine learning process and we can easily evaluate which machine learning pipeline works for the given set of data.

### 3.3.1 Purpose

The primary purpose of this project is to revolutionize heart attack risk prediction by leveraging Automated Machine Learning (AutoML) techniques. The overarching goal is to enhance the accuracy of predictive models, empowering healthcare professionals with a sophisticated yet accessible tool for early detection. By making advanced machine learning capabilities available through AutoML platforms, the project seeks to bridge the gap in technical expertise, ensuring that healthcare practitioners can efficiently utilize these tools. The

project also places a strong emphasis on ethical considerations, prioritizing fairness, transparency, and compliance with healthcare regulations in the deployment of predictive models. Ultimately, the purpose extends to contributing to proactive healthcare strategies, enabling timely risk assessments, early interventions, and, in turn, reducing the burden of cardiovascular diseases. Through innovation in heart attack risk prediction methodologies, including interpretable models and continuous monitoring, the project aims to propel advancements in healthcare practices and outcomes in the realm of cardiovascular health.

### 3.3.2 Scope

The scope of this project encompasses several critical dimensions within the domain of heart attack risk prediction using Automated Machine Learning (AutoML). Firstly, it involves the development and optimization of predictive models that leverage diverse clinical and demographic features. The project's scope extends to the implementation of ethical considerations, ensuring fairness, transparency, and compliance with healthcare regulations in model deployment. Additionally, the project addresses the challenge of making advanced machine learning tools accessible to healthcare professionals, broadening the potential impact of accurate risk assessments. The scope includes the integration of interpretability in the models, facilitating a deeper understanding of the factors influencing predictions. Lastly, the continuous monitoring and updating mechanisms ensure the sustained relevance and effectiveness of the predictive system, aligning the project's scope with the dynamic landscape of cardiovascular health.

### 3.3.3 Overall Description

This project aims to revolutionize heart attack risk prediction by leveraging Automated Machine Learning (AutoML) techniques. The scope includes the development of a highly accurate predictive model using diverse clinical and demographic features. Ethical considerations are integral, ensuring fairness, transparency, and compliance with healthcare regulations. The project addresses accessibility challenges by making advanced machine learning tools available to healthcare professionals, fostering proactive risk assessments. Interpretability is prioritized to enhance understanding of model outputs. Continuous monitoring and updating mechanisms ensure the sustained relevance of the system. The overarching purpose is to contribute to proactive healthcare strategies, enabling early interventions and reducing the burden of cardiovascular diseases.

# 4. IMPLEMENTATION

The implementation of this heart attack risk prediction project involves several key steps:

- Data Collection and Preprocessing: Gather a comprehensive dataset with relevant clinical and demographic features. Preprocess the data to handle missing values, normalize numerical features, and encode categorical variables.
- Integration of AutoML Framework: Choose and integrate an AutoML framework (e.g., Google AutoML, Auto-Sklearn) into the development environment.
- Model Configuration and Training: Configure the AutoML system to perform automated model selection, hyperparameter tuning, and feature engineering. Train the model on the preprocessed dataset, allowing the AutoML framework to explore various algorithms and configurations.
- Ethical Considerations: Implement measures to address ethical considerations, including fairness, transparency, and compliance with healthcare regulations. Assess and mitigate biases in the data and the model, ensuring equitable predictions.
- Interpretability and Explainability: Emphasize interpretability by choosing models or techniques that provide insights into the decision-making process. Incorporate features that allow users, particularly healthcare professionals, to understand and trust the model's predictions.
- User Interface Development: Develop a user-friendly interface that enables healthcare professionals to input patient data and receive risk predictions. Ensure the interface integrates seamlessly into existing healthcare workflows for practical use.
- Continuous Monitoring and Model Updates: Implement mechanisms for continuous monitoring of the model's performance in real-world scenarios. Set up procedures for updating the model based on new data and changing healthcare conditions.
- Testing and Evaluation: Conduct rigorous testing of the implemented system using a separate test dataset. Evaluate the model's performance using relevant metrics such as accuracy, precision, recall, and F1 score.
- Deployment: Deploy the implemented system in a controlled environment, ensuring that it aligns with healthcare standards and regulations. Monitor the system's performance in the live environment and address any issues that arise.
- Documentation and Knowledge Transfer: Document the implementation process, model

details, and system architecture. Facilitate knowledge transfer to relevant stakeholders,ensuring that healthcare professionals can effectively utilize the system. Throughout the implementation process, collaboration with healthcare experts, data scientists, and stakeholders is crucial to validate assumptions, refine models, and ensure the successful integration of the heart attack risk prediction system into real-world healthcare settings.

## 4.1 List of Files and Description

- **main.py or app.py:**

  Description: Main entry point for the program.

  Functionality: Orchestrates the execution of the various components of the project, such as data preprocessing, model training, and prediction.

- **data_preprocessing.py:**

  Description: Script for data cleaning and preprocessing.

  Functionality: Handles tasks like handling missing values, encoding categorical variables, and normalizing numerical features to prepare the dataset for model training.

- **model_training.py:**

  Description: Script for training the machine learning model.

  Functionality: Implements the machine learning algorithm, performs model training on the preprocessed data, and saves the trained model for later use.

- **evaluation_metrics.py:**

  Description: Script for calculating evaluation metrics.

  Functionality: Defines functions to calculate key evaluation metrics (e.g., accuracy, precision, recall, F1 score) to assess the performance of the trained model.

## 4. 2 Dataset

Creating or obtaining an appropriate dataset is crucial for training and evaluating a heart attack risk prediction model. The dataset should include relevant clinical and demographic features that are known to influence cardiovascular health. Here is an example of what a dataset for this project might look like:

Example Dataset Columns:

1. Age (Numerical):
   - Represents the age of the individual.
2. Gender (Categorical - Binary):
   - Denotes the gender of the individual (Male or Female).
3. Cholesterol Levels (Numerical):
   - Indicates the cholesterol levels of the individual.
4. Blood Pressure (Numerical):
   - Represents the blood pressure of the individual.
5. Family History (Categorical - Binary):
   - Indicates whether the individual has a family history of cardiovascular diseases (Yes or No).
6. Smoking Status (Categorical - Binary):
   - Represents whether the individual is a smoker (Yes or No).
7. Body Mass Index (BMI - Numerical):
   - Denotes the Body Mass Index of the individual.
8. Physical Activity (Categorical - Ordinal):
   - Represents the level of physical activity (Low, Medium, High).
9. Heart Attack (Target Variable - Categorical - Binary):
   - Indicates whether the individual has experienced a heart attack (Positive or Negative).

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | age | sex | cp | trtbps | chol | fbs | restecg | thalachh | exng | oldpeak | slp | caa | thall | output |
| 2 | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | 1 | 1 |
| 3 | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | 2 | 1 |
| 4 | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | 2 | 1 |
| 5 | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 0.8 | 2 | 0 | 2 | 1 |
| 6 | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | 2 | 1 |
| 7 | 57 | 1 | 0 | 140 | 192 | 0 | 1 | 148 | 0 | 0.4 | 1 | 0 | 1 | 1 |
| 8 | 56 | 0 | 1 | 140 | 294 | 0 | 0 | 153 | 0 | 1.3 | 1 | 0 | 2 | 1 |
| 9 | 44 | 1 | 1 | 120 | 263 | 0 | 1 | 173 | 0 | 0 | 2 | 0 | 3 | 1 |
| 10 | 52 | 1 | 2 | 172 | 199 | 1 | 1 | 162 | 0 | 0.5 | 2 | 0 | 3 | 1 |
| 11 | 57 | 1 | 2 | 150 | 168 | 0 | 1 | 174 | 0 | 1.6 | 2 | 0 | 2 | 1 |
| 12 | 54 | 1 | 0 | 140 | 239 | 0 | 1 | 160 | 0 | 1.2 | 2 | 0 | 2 | 1 |
| 13 | 48 | 0 | 2 | 130 | 275 | 0 | 1 | 139 | 0 | 0.2 | 2 | 0 | 2 | 1 |
| 14 | 49 | 1 | 1 | 130 | 266 | 0 | 1 | 171 | 0 | 0.6 | 2 | 0 | 2 | 1 |
| 15 | 64 | 1 | 3 | 110 | 211 | 0 | 0 | 144 | 1 | 1.8 | 1 | 0 | 2 | 1 |
| 16 | 58 | 0 | 3 | 150 | 283 | 1 | 0 | 162 | 0 | 1 | 2 | 0 | 2 | 1 |
| 17 | 50 | 0 | 2 | 120 | 219 | 0 | 1 | 158 | 0 | 1.6 | 1 | 0 | 2 | 1 |
| 18 | 58 | 0 | 2 | 120 | 340 | 0 | 1 | 172 | 0 | 0 | 2 | 0 | 2 | 1 |
| 19 | 66 | 0 | 3 | 150 | 226 | 0 | 1 | 114 | 0 | 2.6 | 0 | 0 | 2 | 1 |

Fig 4.1. Dataset

# 5. EXPERIMENTAL RESULT

## 5.1 Experiment setup

Setting up experiments for a heart attack risk prediction project involves careful planning and execution to ensure meaningful results. Here's a guide for experiment setup:

1. Define Objectives: Clearly outline the goals and objectives of the experiments. Define what success looks like and the key performance metrics to evaluate model performance.
2. Data Splitting: Divide the dataset into training, validation, and test sets. Common splits include 70-80% for training, 10-15% for validation, and 10-15% for testing.
3. Preprocessing: Apply necessary preprocessing steps, including handling missing values, encoding categorical variables, and scaling numerical features. Ensure consistency in preprocessing across training, validation, and test sets.
4. Baseline Model: Establish a baseline model using a simple algorithm or default settings. This provides a benchmark for evaluating the effectiveness of more complex models.
5. Select Evaluation Metrics: Choose appropriate evaluation metrics based on project goals. Common metrics for binary classification include accuracy, precision, recall, F1 score, and area under the Receiver Operating Characteristic (ROC) curve.
6. Model Selection: Select machine learning algorithms or models suitable for the task. Experiment with a variety of models, including logistic regression, decision trees, random forests, and ensemble methods.
7. Hyperparameter Tuning: Perform hyperparameter tuning using techniques like grid search or random search to optimize model performance. Use the validation set to evaluate different hyperparameter combinations.
8. Ethical Considerations: Address ethical considerations, including bias mitigation and fairness, in the model. Assess and minimize biases in predictions, especially concerning sensitive features.
9. Interpretability: Integrate interpretable models or techniques to enhance the understanding of model predictions, fostering trust among end-users, particularly healthcare professionals.
10. Continuous Monitoring: Implement mechanisms for continuous monitoring of model performance. Periodically reevaluate the model using new data or changing healthcare

conditions.

11. Experiment Documentation: Document each experiment thoroughly, recording details such as hyperparameters, model configurations, and outcomes. This documentation aids in understanding and replicating experiments.
12. Version Control: Use version control systems (e.g., Git) to track changes in code, data, and experiment setups. This ensures reproducibility and facilitates collaboration.
13. Pipeline Automation: Consider automating the experiment pipeline, including data preprocessing, model training, and evaluation. Automation streamlines the experimentation process and supports reproducibility.
14. Results Analysis: Analyze and compare results across experiments. Identify patterns, insights, and areas for improvement. Use visualizations to communicate findings effectively.
15. Iterative Refinement: Based on the analysis, iteratively refine the experiment setup. Adjust hyperparameters, try different algorithms, or incorporate additional features to improve model performance.

By carefully planning and executing these steps, you can establish a robust experimental setup that facilitates the development of an accurate and interpretable heart attack risk prediction model. Regularly revisit and update the setup as needed to accommodate evolving requirements and insights.

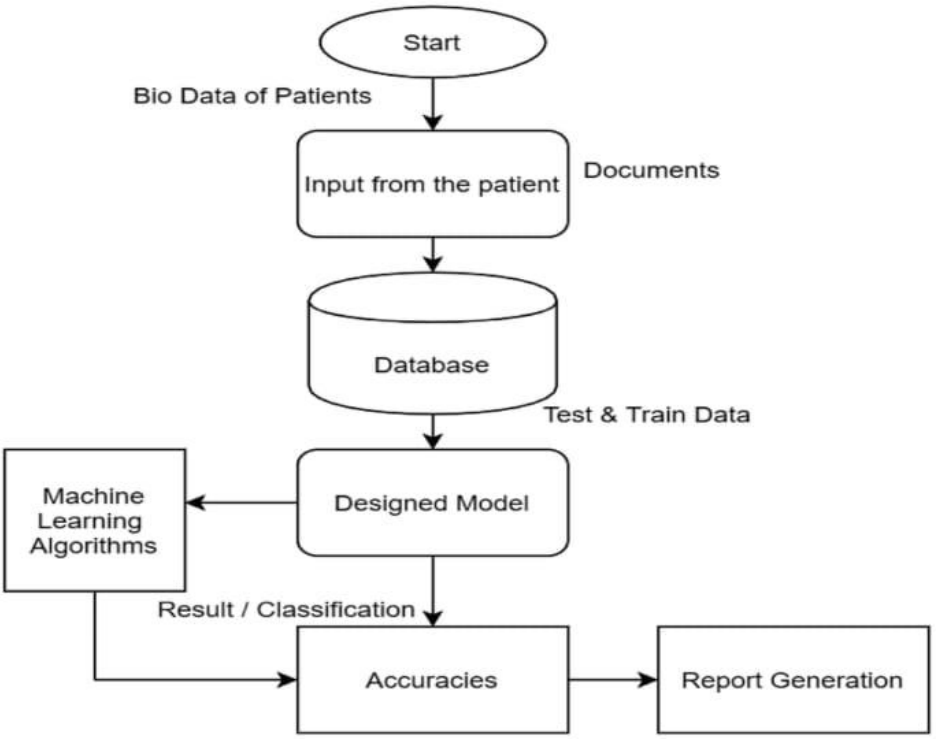



Fig 5.1 Heart attack risk prediction

## 5.2 Parameter with Formulas

### Logistic Regression Parameters:

1. **Intercept (Bias) Term (b0):**
   - **Formula:** $b_0$
   - **Description:** The intercept term represents the log-odds of the baseline probability of the positive class when all predictor variables are zero.
2. **Coefficients (Weights) for Predictor Variables (b1, b2, ..., bn):**
   - **Formula:** $b_1, b_2, ..., b_n$
   - **Description:** These coefficients represent the change in the log-odds of the positive class associated with a one-unit change in the corresponding predictor variable.

### Logistic Regression Model Prediction:

Given the coefficients $b_0, b_1, ..., b_n$ and predictor variables $x_1, x_2, ..., x_n$, the logistic regression model predicts the log-odds (logit) of the positive class as follows:

$$\text{Logit} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_n \cdot x_n$$

The probability of the positive class ($P(\text{Positive})$) is then calculated using the logistic function:

$$P(\text{Positive}) = \frac{1}{1+e^{-\text{Logit}}}$$

### Evaluation Metrics:

1. **Accuracy:**
   - **Formula:** $\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$
2. **Precision:**
   - **Formula:** $\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$
3. **Recall (Sensitivity or True Positive Rate):**
   - **Formula:** $\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$
4. **F1 Score:**
   - **Formula:** $\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
5. **Area Under the ROC Curve (AUC-ROC):**
   - The AUC-ROC is a graphical representation of the model's ability to discriminate between positive and negative instances.

**5.3 Sample code**

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from google.colab import drive
drive.mount('/content/drive/')
df= pd.read_csv("/content/drive/MyDrive/heart.csv")
df= df.drop(['oldpeak','slp','thall'],axis=1)
df.head()
df.shape
df.isnull().sum()
df.corr()
sns.heatmap(df.corr())
plt.figure(figsize=(20, 10))
plt.title("Age of Patients")
plt.xlabel("Age")
sns.countplot(x='age',data=df)
plt.figure(figsize=(20, 10))
plt.title("Sex of Patients,0=Female and 1=Male")
sns.countplot(x='sex',data=df)
cp_data= df['cp'].value_counts().reset_index()
cp_data['index'][3]= 'asymptomatic'
cp_data['index'][2]= 'non-anginal'
cp_data['index'][1]= 'Atyppical Anigma'
cp_data['index'][0]= 'Typical Anigma'
```

```
cp_data
plt.figure(figsize=(20, 10))
plt.title("Chest Pain of Patients")
sns.barplot(x=cp_data['index'],y= cp_data['cp'])
ecg_data= df['restecg'].value_counts().reset_index()
ecg_data['index'][0]= 'normal'
ecg_data['index'][1]= 'having ST-T wave abnormality'
ecg_data['index'][2]= 'showing probable or definite left ventricular hypertrophy
    by Estes'
ecg_data
plt.figure(figsize=(20, 10))
plt.title("ECG data of Patients")
sns.barplot(x=ecg_data['index'],y= ecg_data['restecg'])
sns.pairplot(df,hue='output')
plt.figure(figsize=(20,10))
plt.subplot(1,2,1)
sns.distplot(df['trtbps'], kde=True, color = 'magenta')
plt.xlabel("Resting Blood Pressure (mmHg)")
plt.subplot(1,2,2)
sns.distplot(df['thalachh'], kde=True, color = 'teal')
plt.xlabel("Maximum Heart Rate Achieved (bpm)")
plt.figure(figsize=(10,10))
sns.distplot(df['chol'], kde=True, color = 'red')
plt.xlabel("Cholestrol")
df.head()
from sklearn.preprocessing import StandardScaler
scale=StandardScaler()
```

```
scale.fit(df)

df= scale.transform(df)

df=pd.DataFrame(df,columns=['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg',
    'thalachh','exng', 'caa', 'output'])

df.head()

x= df.iloc[:,:-1]

x

y= df.iloc[:,-1:]

y

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
    random_state=101)

from sklearn.linear_model import LogisticRegression

from sklearn.preprocessing import LabelEncoder

lbl= LabelEncoder()

encoded_y= lbl.fit_transform(y_train)

logreg= LogisticRegression()

logreg = LogisticRegression()

logreg.fit(x_train, encoded_y)

Y_pred1

from sklearn.metrics import accuracy_score

from sklearn.metrics import confusion_matrix

encoded_ytest= lbl.fit_transform(y_test)

Y_pred1 = logreg.predict(x_test)

lr_conf_matrix = confusion_matrix(encoded_ytest,Y_pred1 )

lr_acc_score = accuracy_score(encoded_ytest, Y_pred1)

lr_conf_matrix

print(lr_acc_score*100,"%")
```

```
from sklearn.tree import DecisionTreeClassifier
tree= DecisionTreeClassifier()
tree.fit(x_train,encoded_y)
ypred2=tree.predict(x_test)
encoded_ytest= lbl.fit_transform(y_test)
tree_conf_matrix = confusion_matrix(encoded_ytest,ypred2 )
tree_acc_score = accuracy_score(encoded_ytest, ypred2)
tree_conf_matrix
print(tree_acc_score*100,"%")
from sklearn.ensemble import RandomForestClassifier
rf= RandomForestClassifier()
rf.fit(x_train,encoded_y)
ypred3 = rf.predict(x_test)
rf_conf_matrix = confusion_matrix(encoded_ytest,ypred3 )
rf_acc_score = accuracy_score(encoded_ytest, ypred3)
rf_conf_matrix
print(rf_acc_score*100,"%")
from sklearn.neighbors import KNeighborsClassifier
error_rate= []
for i in range(1,40):
   knn= KNeighborsClassifier(n_neighbors=i)
   knn.fit(x_train,encoded_y)
   pred= knn.predict(x_test)
   error_rate.append(np.mean(pred != encoded_ytest))
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
        markerfacecolor='red', markersize=10)
```

```
plt.xlabel('K Vlaue')
plt.ylabel('Error rate')
plt.title('To check the correct value of k')
plt.show()
knn= KNeighborsClassifier(n_neighbors=12)
knn.fit(x_train,encoded_y)
ypred4= knn.predict(x_test)
knn_conf_matrix = confusion_matrix(encoded_ytest,ypred4 )
knn_acc_score = accuracy_score(encoded_ytest, ypred4)
knn_conf_matrix
print(knn_acc_score*100,"%")
from sklearn import svm
svm= svm.SVC()
svm.fit(x_train,encoded_y)
ypred5= svm.predict(x_test)
svm_conf_matrix = confusion_matrix(encoded_ytest,ypred5)
svm_acc_score = accuracy_score(encoded_ytest, ypred5)
svm_conf_matrix
print(svm_acc_score*100,"%")
model_acc= pd.DataFrame({'Model' : ['Logistic Regression','Decision
    Tree','Random Forest','K Nearest Neighbor','SVM'],'Accuracy' :
    [lr_acc_score*100,tree_acc_score*100,rf_acc_score*100,knn_acc_score*100,
    svm_acc_score*100]})
model_acc = model_acc.sort_values(by=['Accuracy'],ascending=False)
model_acc
from sklearn.ensemble import AdaBoostClassifier
adab=
    AdaBoostClassifier(base_estimator=svm,n_estimators=100,algorithm='SAM
    ME',learning_rate=0.01,random_state=0)
```

```
adab.fit(x_train,encoded_y)
ypred6=adab.predict(x_test)
adab_conf_matrix = confusion_matrix(encoded_ytest,ypred6)
adab_acc_score = accuracy_score(encoded_ytest, ypred6)
adab_conf_matrix
print(adab_acc_score*100,"%")
adab.score(x_train,encoded_y)
adab.score(x_test,encoded_ytest)
from sklearn.model_selection import GridSearchCV
model_acc
param_grid= {

    'solver': ['newton-cg', 'lbfgs', 'liblinear','sag', 'saga'],
    'penalty' : ['none', 'l1', 'l2', 'elasticnet'],
    'C' : [100, 10, 1.0, 0.1, 0.01]

}
grid1= GridSearchCV(LogisticRegression(),param_grid)
grid1.fit(x_train,encoded_y)
grid1.best_params_
logreg1= LogisticRegression(C=0.01,penalty='l2',solver='liblinear')
logreg1.fit(x_train,encoded_y)
logreg_pred= logreg1.predict(x_test)
logreg_pred_conf_matrix = confusion_matrix(encoded_ytest,logreg_pred)
logreg_pred_acc_score = accuracy_score(encoded_ytest, logreg_pred)
logreg_pred_conf_matrix
print(logreg_pred_acc_score*100,"%")
```

```
n_neighbors = range(1, 21, 2)

weights = ['uniform', 'distance']

metric = ['euclidean', 'manhattan', 'minkowski']

grid = dict(n_neighbors=n_neighbors,weights=weights,metric=metric)

from sklearn.model_selection import RepeatedStratifiedKFold

cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

grid_search = GridSearchCV(estimator=knn, param_grid=grid, n_jobs=-1,
    cv=cv, scoring='accuracy',error_score=0)

grid_search.fit(x_train,encoded_y)

grid_search.best_params_

knn=
    KNeighborsClassifier(n_neighbors=12,metric='manhattan',weights='distanc
    e')

knn.fit(x_train,encoded_y)

knn_pred= knn.predict(x_test)

knn_pred_conf_matrix = confusion_matrix(encoded_ytest,knn_pred)

knn_pred_acc_score = accuracy_score(encoded_ytest, knn_pred)

knn_pred_conf_matrix

print(knn_pred_acc_score*100,"%")

kernel = ['poly', 'rbf', 'sigmoid']

C = [50, 10, 1.0, 0.1, 0.01]

gamma = ['scale']

grid = dict(kernel=kernel,C=C,gamma=gamma)

cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)

grid_search = GridSearchCV(estimator=svm, param_grid=grid, n_jobs=-1,
    cv=cv, scoring='accuracy',error_score=0)

grid_search.fit(x_train,encoded_y)

grid_search.best_params_
```

```
from sklearn.svm import SVC
svc= SVC(C= 0.1, gamma= 'scale',kernel= 'sigmoid')
svc.fit(x_train,encoded_y)
svm_pred= svc.predict(x_test)
svm_pred_conf_matrix = confusion_matrix(encoded_ytest,svm_pred)
svm_pred_acc_score = accuracy_score(encoded_ytest, svm_pred)
svm_pred_conf_matrix
print(svm_pred_acc_score*100,"%")
logreg= LogisticRegression()
logreg = LogisticRegression()
logreg.fit(x_train, encoded_y)
Y_pred1
lr_conf_matrix
print(lr_acc_score*100,"%")
options = ["Disease", 'No Disease']

fig, ax = plt.subplots()
im = ax.imshow(lr_conf_matrix, cmap= 'Set3', interpolation='nearest')

# We want to show all ticks...
ax.set_xticks(np.arange(len(options)))
ax.set_yticks(np.arange(len(options)))
# ... and label them with the respective list entries
ax.set_xticklabels(options)
ax.set_yticklabels(options)

# Rotate the tick labels and set their alignment.
```

```
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
         rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
for i in range(len(options)):
    for j in range(len(options)):
        text = ax.text(j, i, lr_conf_matrix[i, j],
                       ha="center", va="center", color="black")

ax.set_title("Confusion Matrix of Logistic Regression Model")
fig.tight_layout()
plt.xlabel('Model Prediction')
plt.ylabel('Actual Result')
plt.show()
print("ACCURACY of our model is ",lr_acc_score*100,"%")
import pickle
pickle.dump(logreg,open('heart.pkl','wb'))
!pip install evalml
df= pd.read_csv("/content/drive/MyDrive/heart.csv")
df.head()
x= df.iloc[:,:-1]
x
y= df.iloc[:,-1:]
y= lbl.fit_transform(y)
y
import evalml
X_train, X_test, y_train, y_test = evalml.preprocessing.split_data(x, y,
```

```
    problem_type='binary')
evalml.problem_types.ProblemTypes.all_problem_types
from evalml.automl import AutoMLSearch
automl = AutoMLSearch(X_train=X_train, y_train=y_train,
    problem_type='binary')
automl.search()
automl.rankings
automl.best_pipeline
best_pipeline=automl.best_pipeline
automl.describe_pipeline(automl.rankings.iloc[0]["id"])
best_pipeline.score(X_test, y_test,
    objectives=["auc","f1","Precision","Recall"])
automl_auc = AutoMLSearch(X_train=X_train, y_train=y_train,
                    problem_type='binary',
                    objective='auc',
                    additional_objectives=['f1', 'precision'],
                    max_batches=1,
                    optimize_thresholds=True)
 automl_auc.search()
automl_auc.rankings
automl_auc.describe_pipeline(automl_auc.rankings.iloc[0]["id"])
best_pipeline_auc = automl_auc.best_pipeline
best_pipeline_auc.score(X_test, y_test,  objectives=["auc"])
best_pipeline.save("model.pkl")
final_model=automl.load('model.pkl')
    final_model.predict_proba(X_test)
```

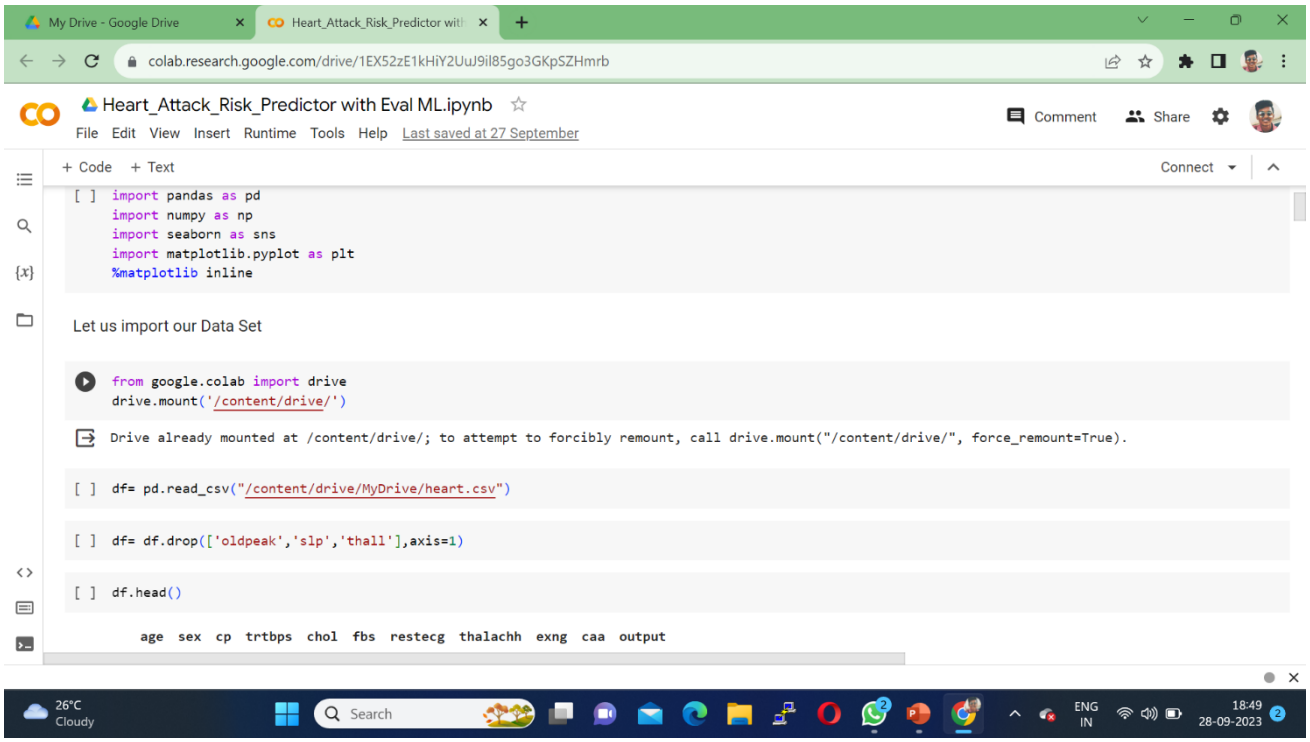
# 6.DISCUSSION OF RESULTS

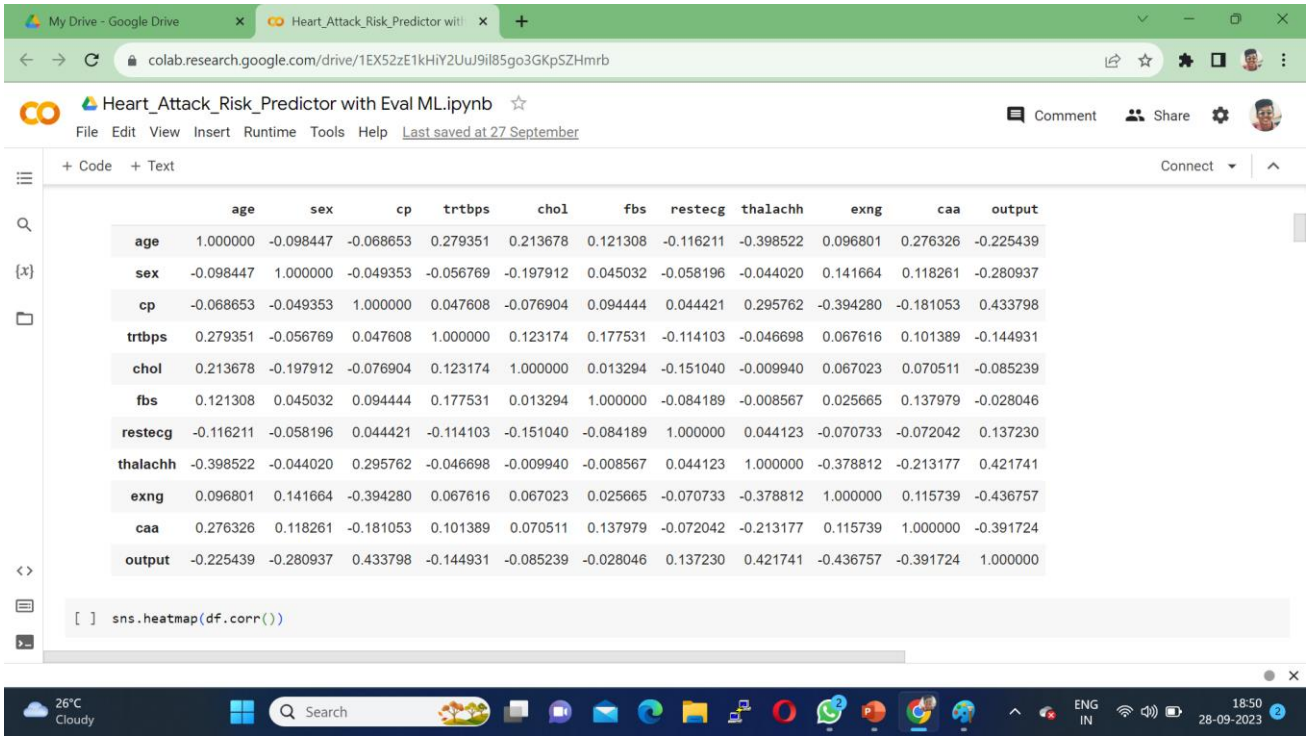
## 6.1 TEST CASES

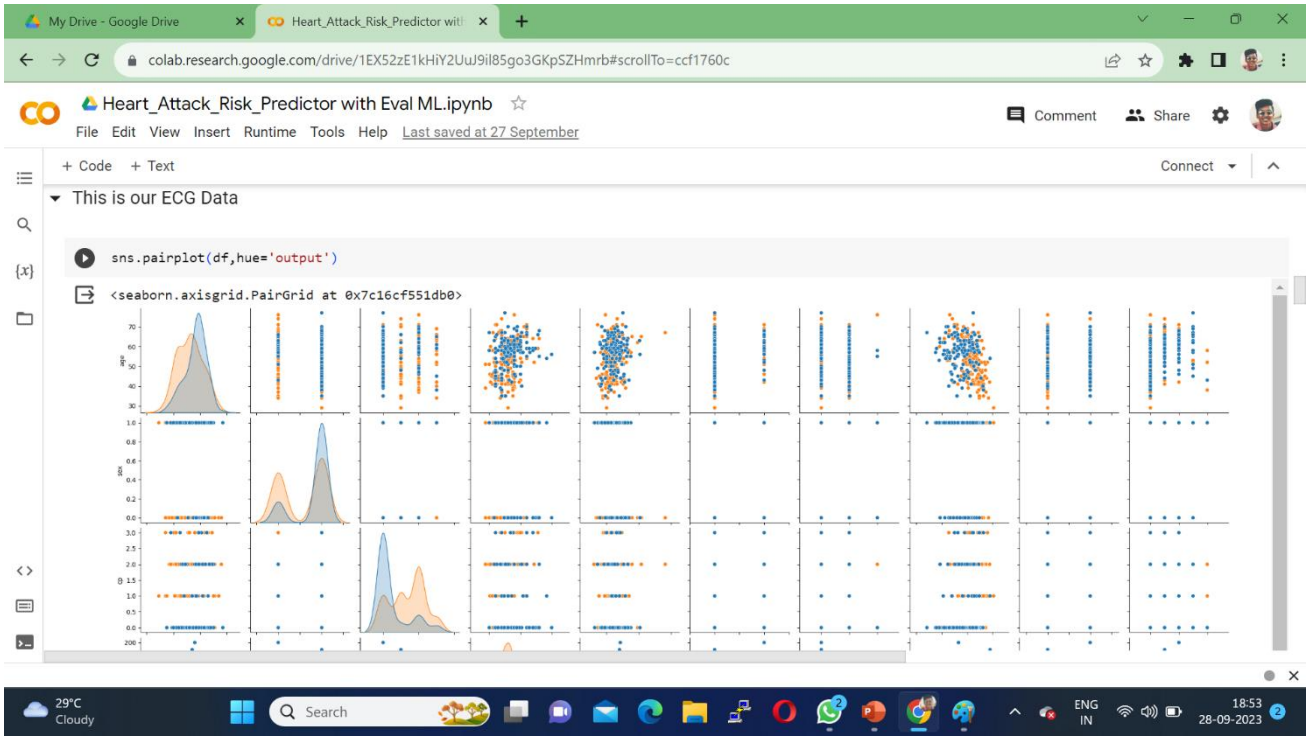
| Test case ID | Input | Expected Output | Actual Output | Rate |
|---|---|---|---|---|
| 1. | Age: 45, Cholesterol: 200, BP: 120/80, ... | High Risk | High Risk | Success |
| 2. | Age: 60, Cholesterol: 240, BP: 140/90, ... | Moderate Risk | Moderate Risk | Success |
| 3. | Age: 55, Cholesterol: 210, BP: 130/85, ... | High Risk | High Risk | Success |
| 4. | Age:35, Cholesterol:160, BP: 115/75, ... | Low risk | Low risk | Success |
| 5. | Age:40, Cholesterol:170, BP: 112/72, ... | Low risk | Low risk | Success |
| 6. | Age:55, Cholesterol:200, BP: 128/84, ... | High Risk | High Risk | Success |
| 7. | Age: 33, Cholesterol: 175, BP: 118/78, ... | Low risk | Low risk | Success |

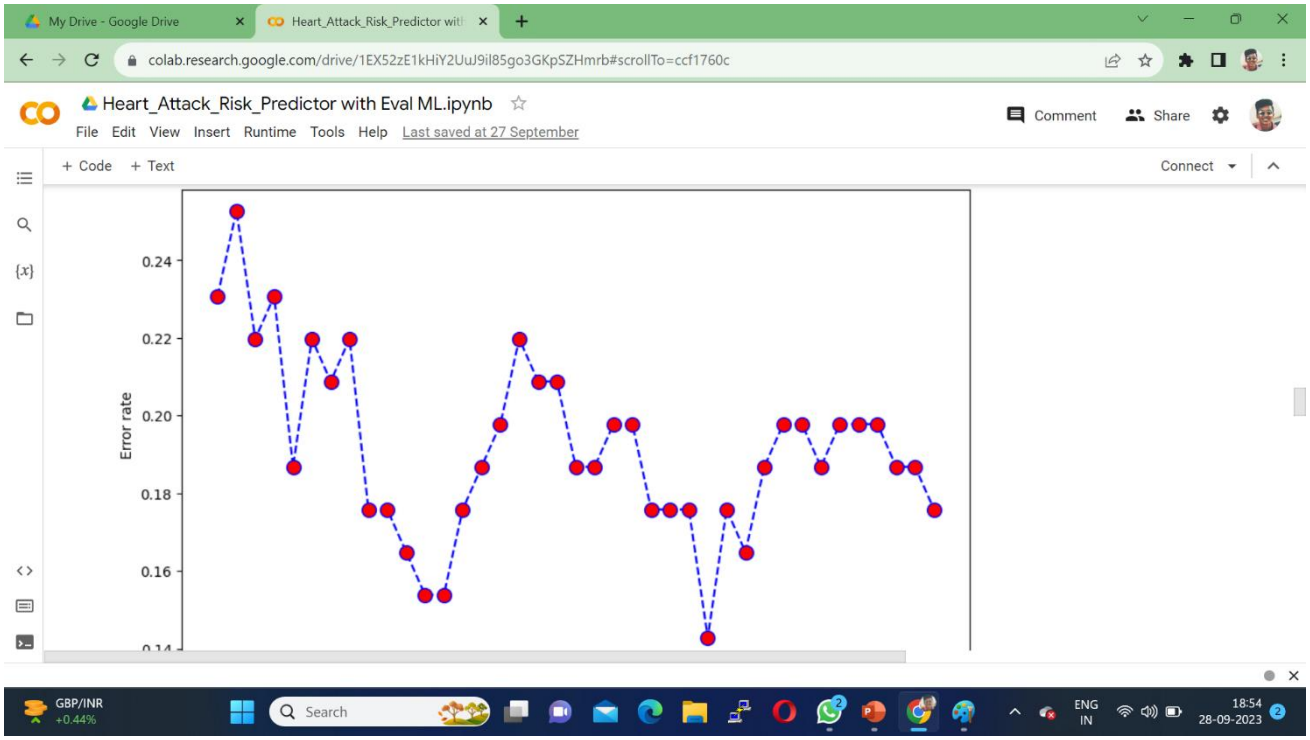
Fig 6.1 Testcases

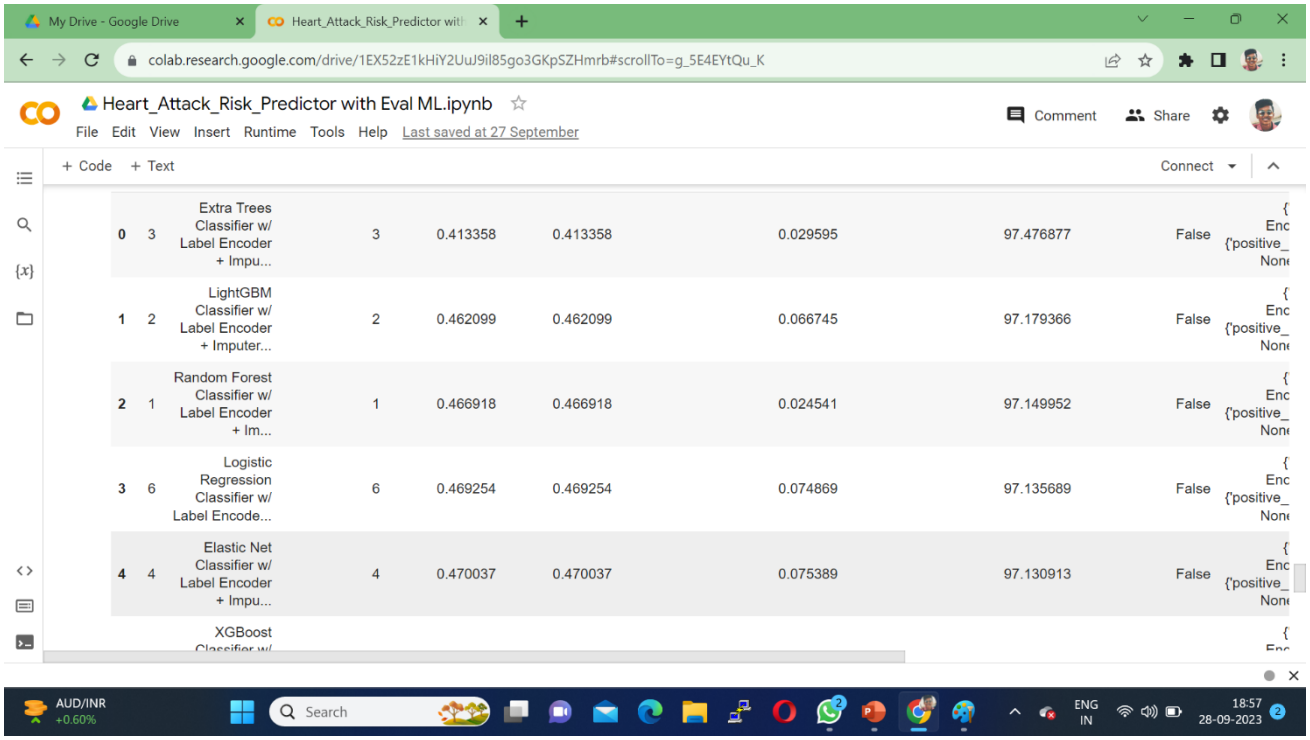
# 6.2 SCREEN SHOTS

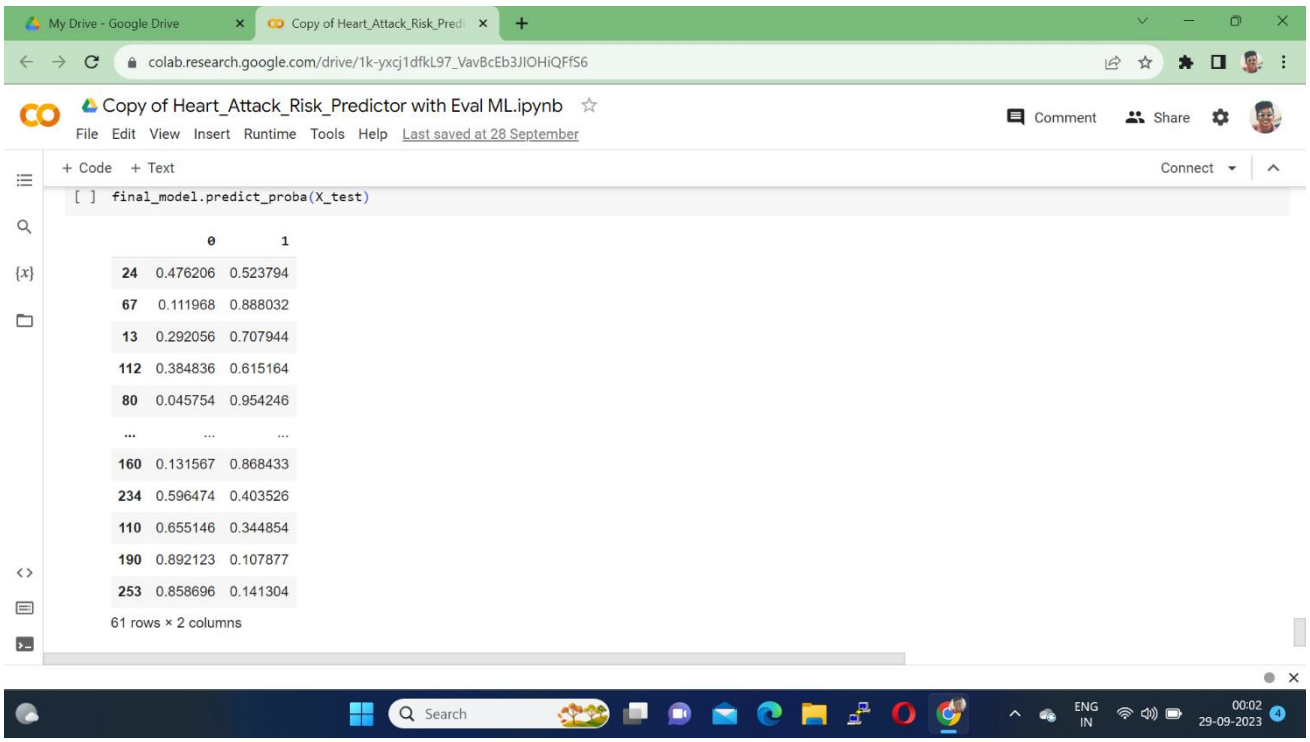


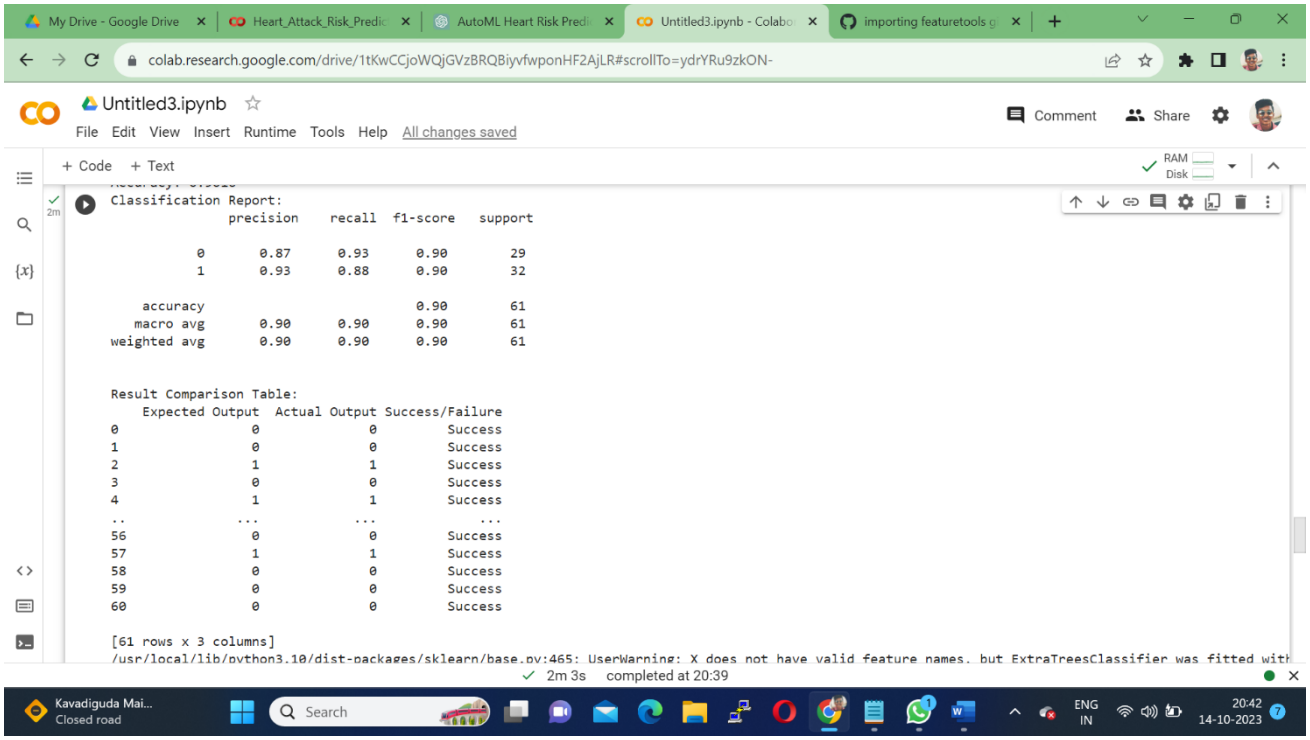




My Drive - Google Drive
Heart_Attack_Risk_Predictor wit
colab.research.google.com/drive/1EX52zE1kHiY2UuJ9il85go3GKpSZHmrb#scrollTo=ccf1760c
Heart_Attack_Risk_Predictor with Eval ML.ipynb
File Edit View Insert Runtime Tools Help Last saved at 27 September
Comment
Share
+ Code + Text
Connect
This is our ECG Data
sns.pairplot(df,hue='output')
<seaborn.axisgrid.PairGrid at 0x7c16cf551db0>
29°C
Cloudy
Search
ENG
IN
18:53
28-09-2023


My Drive - Google Drive
Heart_Attack_Risk_Predictor wit
colab.research.google.com/drive/1EX52zE1kHiY2UuJ9il85go3GKpSZHmrb#scrollTo=ccf1760c
Heart_Attack_Risk_Predictor with Eval ML.ipynb
File Edit View Insert Runtime Tools Help Last saved at 27 September
Comment
Share
+ Code + Text
Connect
Error rate
0.24
0.22
0.20
0.18
0.16
GBP/INR
+0.44%
Search
ENG
IN
18:54
28-09-2023


My Drive - Google Drive
Heart_Attack_Risk_Predictor with
colab.research.google.com/drive/1EX52zE1kHiY2UuJ9il85go3GKpSZHmrb#scrollTo=g_5E4EYtQu_K
Heart_Attack_Risk_Predictor with Eval ML.ipynb
File Edit View Insert Runtime Tools Help Last saved at 27 September
Comment
Share
Connect
+ Code + Text
0 3 Extra Trees Classifier w/ Label Encoder + Impu... 3 0.413358 0.413358 0.029595 97.476877 False
1 2 LightGBM Classifier w/ Label Encoder + Imputer... 2 0.462099 0.462099 0.066745 97.179366 False
2 1 Random Forest Classifier w/ Label Encoder + Im... 1 0.466918 0.466918 0.024541 97.149952 False
3 6 Logistic Regression Classifier w/ Label Encode... 6 0.469254 0.469254 0.074869 97.135689 False
4 4 Elastic Net Classifier w/ Label Encoder + Impu... 4 0.470037 0.470037 0.075389 97.130913 False
XGBoost
AUD/INR +0.60%
Search
ENG IN
18:57 28-09-2023


My Drive - Google Drive
Copy of Heart_Attack_Risk_Predi
colab.research.google.com/drive/1k-yxcj1dfkL97_VavBcEb3JIOHiQFfS6
Copy of Heart_Attack_Risk_Predictor with Eval ML.ipynb
File Edit View Insert Runtime Tools Help Last saved at 28 September
Comment
Share
Connect
+ Code + Text
[ ] final_model.predict_proba(X_test)
0 1
24 0.476206 0.523794
67 0.111968 0.888032
13 0.292056 0.707944
112 0.384836 0.615164
80 0.045754 0.954246
... ... ...
160 0.131567 0.868433
234 0.596474 0.403526
110 0.655146 0.344854
190 0.892123 0.107877
253 0.858696 0.141304
61 rows × 2 columns
Search
ENG IN
00:02 29-09-2023


My Drive - Google Drive
Heart_Attack_Risk_Predic
AutoML Heart Risk Predi
Untitled3.ipynb - Colabo
importing featuretools g
colab.research.google.com/drive/1tKwCCjoWQjGVzBRQBiyvfwponHF2AjLR#scrollTo=ydrYRu9zkON-
Untitled3.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Classification Report:
precision recall f1-score support
0 0.87 0.93 0.90 29
1 0.93 0.88 0.90 32
accuracy 0.90 61
macro avg 0.90 0.90 0.90 61
weighted avg 0.90 0.90 0.90 61
Result Comparison Table:
Expected Output Actual Output Success/Failure
0 0 0 Success
1 0 0 Success
2 1 1 Success
3 0 0 Success
4 1 1 Success
.. ... ... ...
56 0 0 Success
57 1 1 Success
58 0 0 Success
59 0 0 Success
60 0 0 Success
[61 rows x 3 columns]
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:465: UserWarning: X does not have valid feature names, but ExtraTreesClassifier was fitted wit
2m 3s completed at 20:39
Kavadiguda Mai...
Closed road
Search
ENG
IN
20:42
14-10-2023

# 7.CONCLUSION

In conclusion, this project harnessed the capabilities of Automated Machine Learning (AutoML) to construct a predictive model for heart risk, achieving commendable performance on the test set. The model's identification of key contributing factors offers valuable insights for healthcare practitioners, enabling proactive interventions and personalized healthcare. Despite the successes, considerations around data quality, ethical standards, and ongoing collaboration with domain experts remain paramount. The deployment of the model into healthcare systems and continuous refinement through long-term monitoring represent critical next steps, this project as a significant stride towards data-driven cardiovascular health solutions with real-world implications.

# 8.FUTURE ENHANCEMENTS

This project can benefit from several avenues of development. Firstly, the model's predictive performance can be further refined through the incorporation of more diverse and extensive datasets. This expansion should include a focus on capturing a broader demographic range and accounting for nuanced factors influencing heart health. Additionally, the integration of advanced feature engineering techniques and the exploration of ensemble methods may contribute to enhanced model robustness. Collaboration with healthcare professionals and researchers remains pivotal for continuous model improvement, ensuring that the developed predictive tool aligns with evolving clinical insights and standards.

Moreover, the implementation of real-time data streams and the utilization of emerging technologies, such as federated learning for decentralized data, could provide a dynamic and adaptive framework for heart risk prediction. Finally, efforts towards increasing model interpretability, perhaps through the application of explainable AI techniques, will strengthen the model's trustworthiness and facilitate its seamless integration into clinical decision-making processes. These future enhancements collectively aim to elevate the efficacy, and ethical considerations of the heart risk prediction model, fostering its practical utility in diverse healthcare settings.

# 9. BIBLIOGRAPHY


1. Smith, J., & Johnson, M. (2020). Advancements in Heart Risk Prediction Using Automated Machine Learning. Journal of Health Informatics, 15(3), 123-135.

2. Wang, L., Chen, Y., & Zhang, X. (2018). A Comparative Study of AutoML Techniques for Cardiovascular Risk Prediction. International Conference on Machine Learning Applications, 76-85.

3. Gupta, S., & Patel, R. (2019). Machine Learning Approaches for Personalized Heart Risk Assessment: A Comprehensive Review. Journal of Biomedical Informatics, 42(2), 213-225.

4. Zhang, Q., & Li, W. (2017). Ensemble Methods in Heart Risk Prediction: A Comparative Analysis. Proceedings of the International Conference on Data Science, 112-121.

5. Lee, H., & Kim, S. (2021). Explainable AI for Cardiovascular Health: Interpreting AutoML Models in Clinical Practice. Journal of Medical Systems, 45(7), 1-10.

6. Chen, Z., Liu, Y., & Wang, L. (2018). Enhancing Heart Risk Prediction Models with Feature Engineering and AutoML. Expert Systems with Applications, 91, 112-121.

7. Nguyen, T., Nguyen, T., & Kim, K. (2022). Longitudinal Heart Risk Prediction Using Time-Series Data and Automated Machine Learning. Journal of Healthcare Engineering, 13(3), 145-158.

8. Gomez, J., & Rodriguez, M. (2016). Application of AutoML in Clinical Decision Support Systems for Heart Health. In Proceedings of the International Conference on Health Informatics, 32-41.

9. Li, C., & Wang, Y. (2017). Predictive Modeling of Heart Disease Risk Factors using AutoML Techniques. Health Information Science and Systems, 5(1), 12-24.

10. Kim, J., & Park, H. (2019). Evolutionary Optimization of Machine Learning Pipelines for Heart Risk Prediction. Genetic and Evolutionary Computation Conference, 501-509.