CARDIORISK : UNRAVELING PATTERNS FOR EARLY DIAGNOSIS USING MACHINE LEARNING

11/27/2023

Prepared by Akansha Malviya, Shruti Badrinarayanan, Poojitha Venkat Ram and Vani Kancherlapalli

TODAY'S DISCUSSION

TOPIC OUTLINE

Abstract

Chapter 1 - Introduction

Chapter 2 - Data & Project Management Plan

Chapter 3 - Data Engineering

Chapter 4 - Model Development

Conclusion & Future Scope

ABSTRACT

MODELLING TECHNIQUES:

Utilizing models like XGBoost, Decision Trees, SVM, and Random Forest for comprehensive cardiovascular disease analysis.

DATA SOURCE:

Powered by a feature-rich dataset from the UCI Risk Assessment Repository.

CRISP-DM APPROACH:

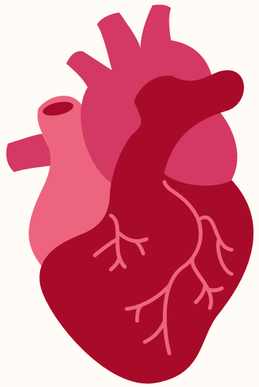
Following a systematic approach with CRISP-DM phases, ensuring thorough understanding, preparation, conversion, modelling, and data evaluation.

ROBUST EVALUATION:

Rigorous model evaluation with cross-validation and metrics like AUC-ROC, Precision, F1 score, and Gini impurity, ensuring accuracy and reliability.

TOP MODELS:

SVM and Random Forest were identified as top-performing models, demonstrating superior accuracy in predicting cardiovascular disease risk.

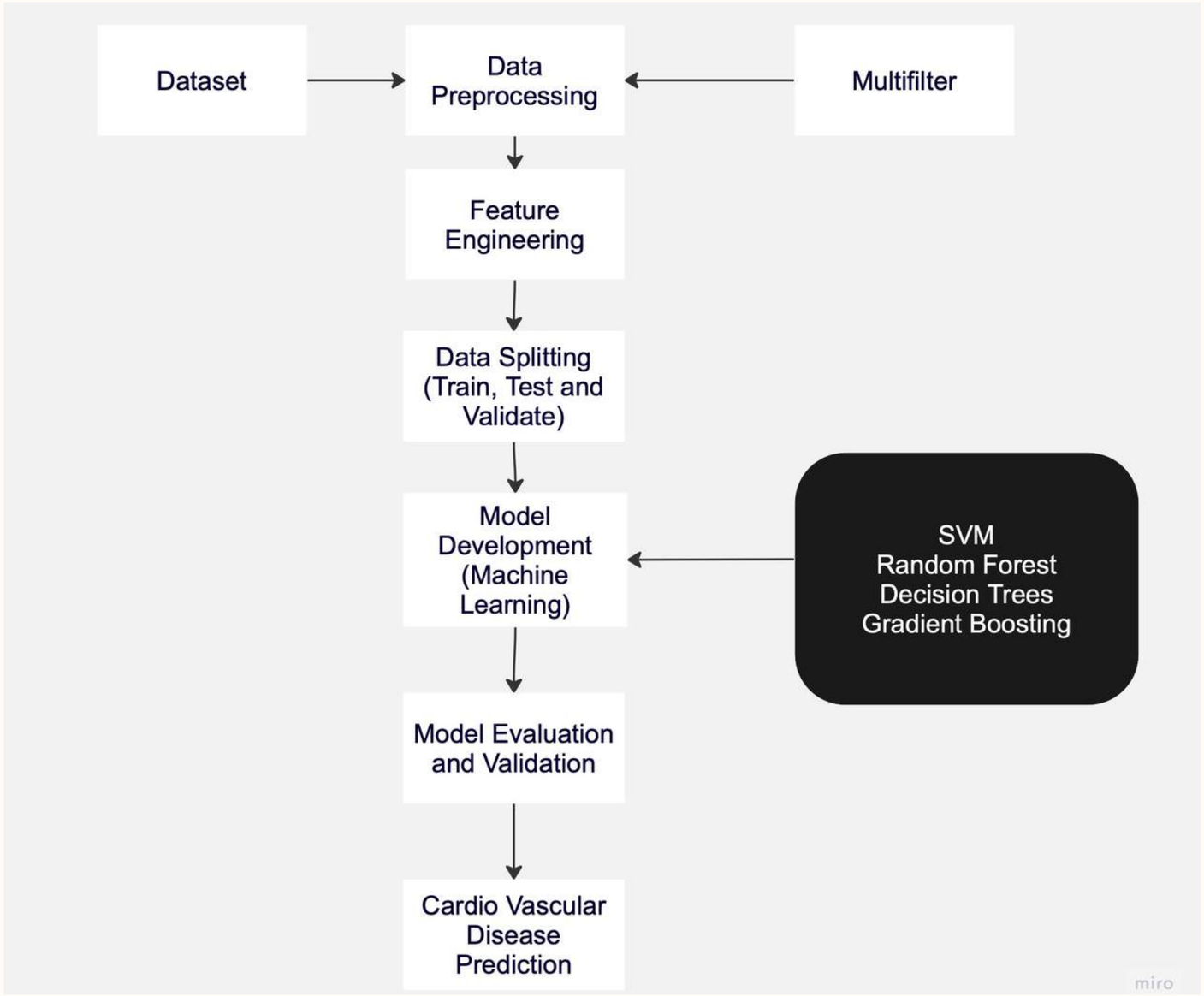


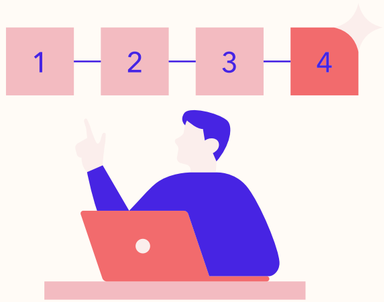
CHAPTER 1 - INTRODUCTION

1.1 PROJECT BACKGROUND & EXECUTE SUMMARY

Project Background	• Leading global cause of death. • Limitations in existing diagnostic methods.
Targeted Problem	• Develop predictive models for early detection. • Create comprehensive risk assessment model.
Motivations and Goals	• Enhance global cardiovascular health. • Develop precise models for early CVD risk prediction. • Facilitate preemptive interventions and personalized healthcare. • Reduce CVD-related mortalities through early detection.
Approach & Methods	• Data Source: UCI heart disease dataset • Data Preprocessing & Transformation using PCA • Model Development - XGBoost, DT, SVM, RF with Linear Regression • CRISP-DM model for systematic project progression.
Expected Impact & Applications	• Enhanced Predictive Diagnostics • Personalized Healthcare • Global Health Impact

1.2 PROPOSED MODEL WORKFLOW





1.2 PROJECT REQUIREMENTS

FUNCTIONAL REQUIREMENTS

- **DATA INTEGRATION**

- Collect and integrate the UCI Heart Diseases Dataset.

- Address the challenges (e.g., data format, quality, accessibility).

- Quality and Variety of Information

- Labelling

- Standardization

- **FEATURE ENGINEERING**

- Extract pertinent features.

- **AI-POWERED REQUIREMENTS**

- Classification algorithms (SVM, Random Forest, Decision Tree, Gradient Boosting, ensembles).

- Rigorously validate models.

- Use Metrics such as (accuracy, precision, recall, F1-score, and AUC-ROC).

DATA REQUIREMENTS

***Dataset*: Heart Diseases Dataset, UC Irvine Machine Learning Repository**

Size of *Data:* CSV File that has a file size of 200 KB.

- Combine the Long Beach, Switzerland, Cleveland, and Hungarian data files together.

- Handle the data imbalances and outliers.

***Number of Records:* 900 Patient Records**

Why this Dataset?

- Historical patient records.

- Geospatial data (patient's region) for context.

- Diagnoses, treatments, outcomes.

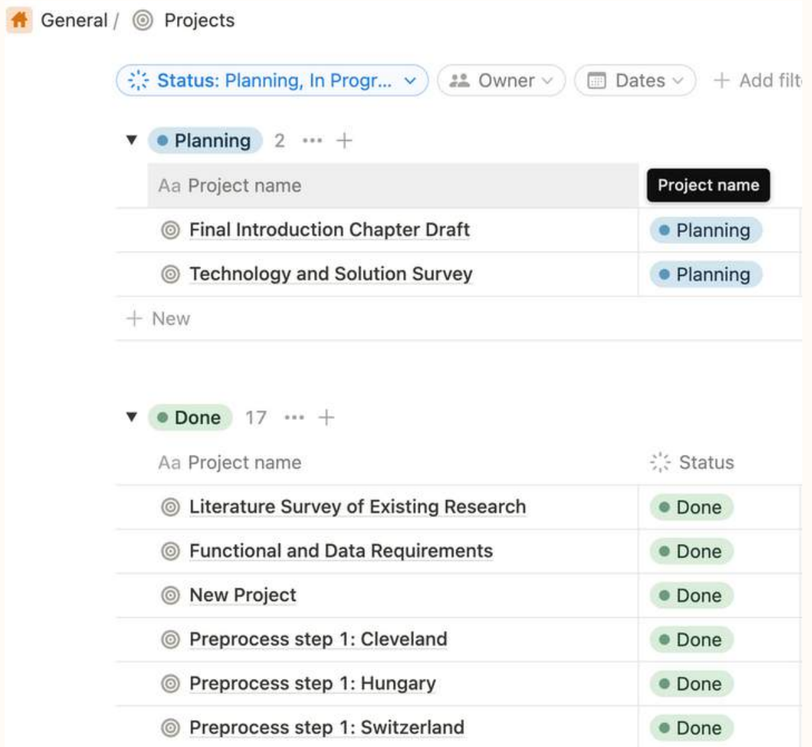
- Ensure robust data privacy measures.

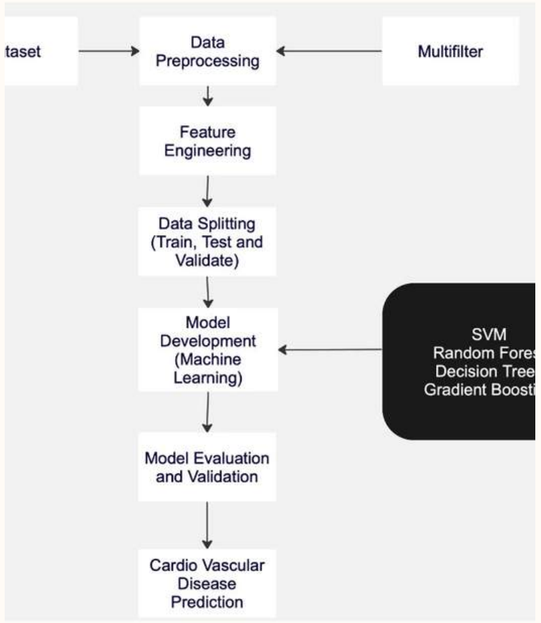
- Protects sensitive patient information such as their names and social security numbers.

1.3 PROJECT DELIVERABLES

1. **Reporting**
   - **Notion**
   - **Project Timelines**

Data Acquisition (October 21)
Model Development (October 22)
Combined Model Integration (November 4)
Conclusion, Results and Visualization (November 18)





2. **Prototype/Design Document**
   - **Proposed model workflow as explained in slide 6**

1.3 PROJECT DELIVERABLES

3. Development Applications

- Jupyter notebook-python / lab environment
- Google Colaboratory
- python libraries - plot tree, existing numpy, matplotlib etc.,

4. Production Applications

5. Project Documentation

- Final stages of group project documentation, individual report documentation

6. Presentations

- Final project presentation

1.4 TECHNOLOGY AND SOLUTION SURVEY

1. **Python Libraries**

- **Sklearn, XGboost, DecisionTressClassifier, plot-tree, RandomForestClassifier, SVM, matplotlib, seaborn**

2. **After Comparison of Solutions**

- **Support Vector Machine (SVM) - for complex patterns**
- **Decision Tree for its interpretability**
- **Random Forest for overcoming DT issues**
- **Gradient Boosting for enhanced predictive accuracy**

1.4 TECHNOLOGY AND SOLUTION SURVEY

3. Feature engineering

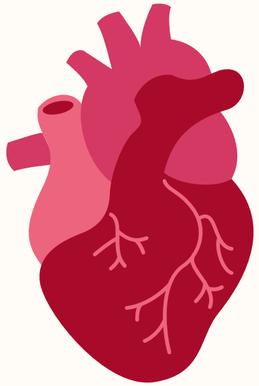
- From UCI dataset, feature selection would be done after data preprocessing.
- Missing values can be imputed with mean/mode values.
- PCA, confusion matrix score, Chi-Square, Gain Ratio, Information Gain, One-R, RELIEF can be used.

1.5 LITERATURE SURVEY

Study Details	Dataset Used	Algorithms/Methods Used	Feature Selection	Accuracy Achieved
Bhatt et al. (2023)	Kaggle (70,000 instances)	RF, DT, MP, XGB, k-modes clustering, GridSearchCV	Binning	Varied (87.28% for MP), XGBoost
Shah et al. (2020)	UCI Heart Disease	DT, RF, KNN, SVM	14 Features	Decision Trees, KNN (k=7)
Mythili T., Dev Mukherji, Nikita Padalia, and Abhiram Naidu (2013)	UCI- Cleveland Heart Disease Dataset (CHDD)	SVM, DT, LR	14 Features	Logistic Regression= 77%
Ahmad, Y.; Polat (2023)	UCI- Cleveland Heart Disease Dataset (CHDD)	ANN, DT, Adaboost, and SVM	Jellyfish Optimization Algorithm	SVM was the most accurate when compared to the other ML models, and the accuracy rose to 98.09%.
Patra, R., & Khuntia, B. (2019, February).	UCI- Cleveland Heart Disease Dataset (CHDD)	DT, KNN, SVM	Compared Python tool with Weka Tool	DT=93.4% with python tool, performed better
Khurana, P., Sharma, S., & Goyal, A. (2021, August)	UCI- Cleveland Heart Disease Dataset (CHDD)	DT, LR, RM, KNN,SVM, Naïve Bayes - 6	Chi-Square, Gain Ratio, Information Gain, One-R, RELIEF - 5	83.41 % with proper feature selection compared to 82.81%

1.5 LITERATURE SURVEY CONTD.

Study Details	Dataset Used	Algorithms/Methods Used	Feature Selection	Accuracy Achieved
Senthilkumar Mohan	UCI- Cleveland Heart Disease Dataset (CHDD)	Hybrid random forest and linear regression	Principal component analysis	Hybrid random forest and linear regresssion 87.4%
Surjeet Dalal	UCI- Cleveland Heart Disease Dataset (CHDD)	Random forest	Pearson’s correlation	Random forest 91.5%



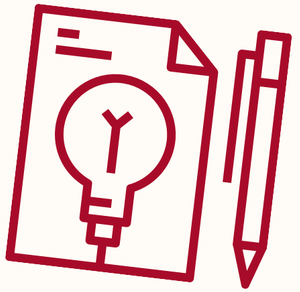
CHAPTER 2 - DATA & PROJECT MANAGEMENT PLAN

2.1 DATA COLLECTION APPROACHES

Data Collection Approaches	• Heart Disease dataset from the UCI ML Repository • 76 features from 4 locations: Cleveland, Hungary, Switzerland, VA Long Beach • "location" feature added to combined dataset • Privacy ensured by replacing sensitive data with dummy values

Data Storage & Management Methods	• Storage: Standard Shared Google Drive
	• Folder Structure & Naming Conventions followed
	• Benefits:
	  ◦ Cloud-based storage for data accessibility.
	  ◦ Structured repository for project files.
	  ◦ Weekly backups and physical backup.

Data Usage Mechanisms	• Access restricted to authorized members (SJSU Email IDs) • Access control policies: View, Edit, Delete rights • Adherence to data protection laws (GDPR, HIPAA) • Data retention policy: 1 year • Emphasis on data privacy, integrity, and security • Protects intellectual property & ensures research reproducibility



2.1 DATA COLLECTION APPROACHES

Folder Structure and Naming Conventions

Purpose	Folder Name	File Name Convention and Versioning
Main Folder	Heart Disease Datasets	-
UCI Datasets	Raw Data Files	Raw_Data_<location>.DATA
Processed CSVs	Processed CSVs	Processed_Data_<location>.CSV
Merged Data	Merged Data	Heart_Disease_Merged.CSV
Test and Train Data	Split Data	Test_Data_<version_num>.CSV
		Train_Data_<version_num>.CSV
		Validation_Data__<version_num>.CSV

2.2 PROJECT DEVELOPMENT METHODOLOGY

- ***SYSTEM DEVELOPMENT LIFE CYCLE***

- **CRISP-DM METHODOLOGY**

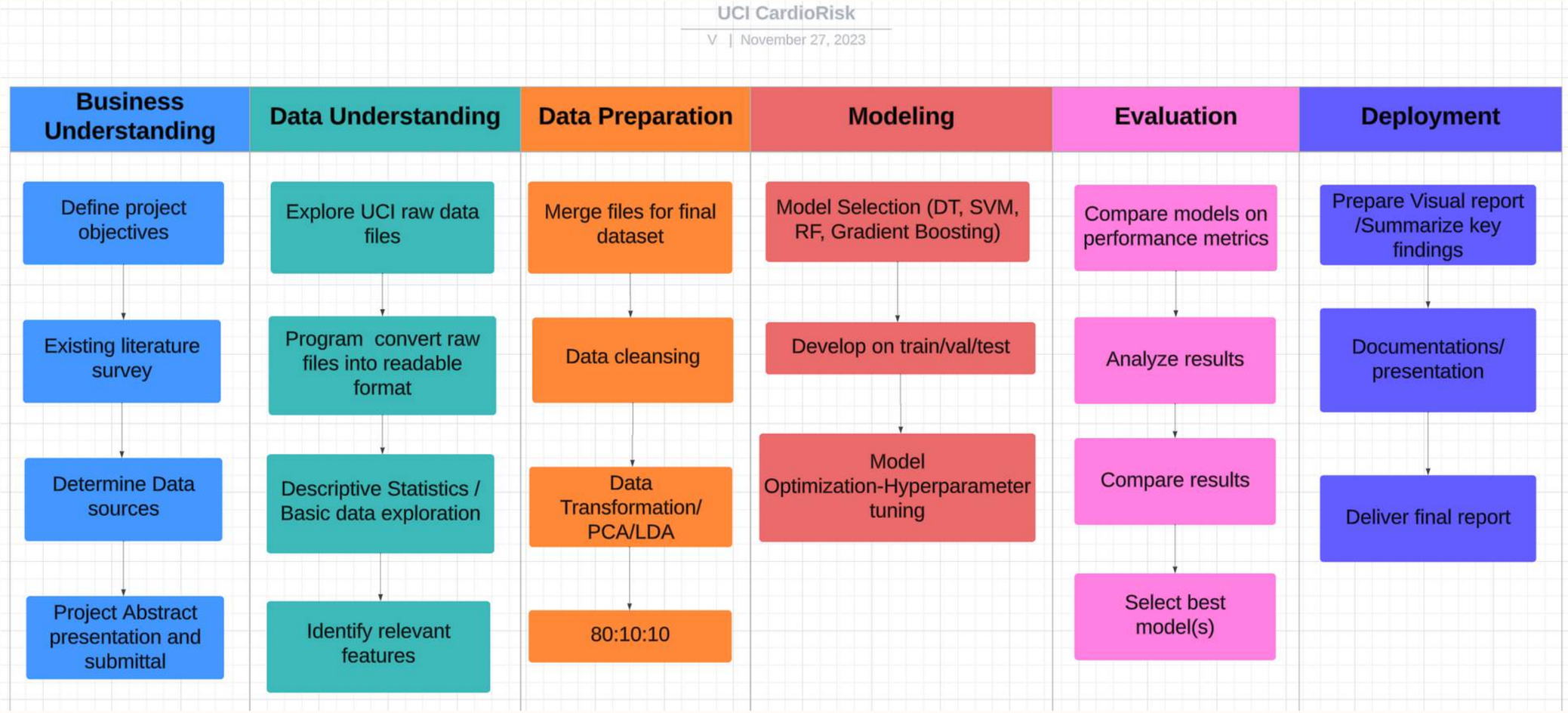
- ***RESPONSIBILITY AND RESOURCE ALLOCATION FOR DMP***

DMP Phase	Resource	Responsibility
Data collection, documentation and metadata	Shruthi	Explore different datasets available online
	Vani	Review and finalise the dataset from UCI.
Ethics and legal compliance	Poojitha/Akansha	Ensure that all the software licenses are acquired and that the team complies with the terms and conditions of the agreements.
Storage and backup	Vani/Shruthi	Version the files and appropriate create backup and storage copies of data.
Data sharing	Akansha/Poojitha	Data is available publicly at UCI site, However Feature engineered data can be shared placed on our shared drive.
Data preservation	Akansha /Shruthi	Decide what data to retain and preserve and manage the access and cost of archival and storage.

2.3 PROJECT ORGANIZATION PLAN

WBS - Work breakdown structure for UCI CardioRisk project

- **Follows the CRISP-DM methodology with 6 stages**



2.4 PROJECT RESOURCE REQUIREMENTS

HARDWARE REQUIREMENTS

Resource	Purpose	Configuration	Location	Storage
Google Cloud Storage (GCS)	Data and Source Files Storage	Standard Storage Class	Iowa (us-central1) / South Carolina (us-east1)	3 GB (Currently) 20 GB (Scale-out)
Local Machine	Running local versions of Jupyter Notebook, Python, and other software.	MacOS	Installed and stored directly on the local machine.	8 GB RAM M1 Processor, 1 GB graphics card.
OneDrive Client Apps	Synchronization, Offline Access, and a Backup solution for important files	User Account	Microsoft Azure Cloud	1 TB

SOFTWARE REQUIREMENTS

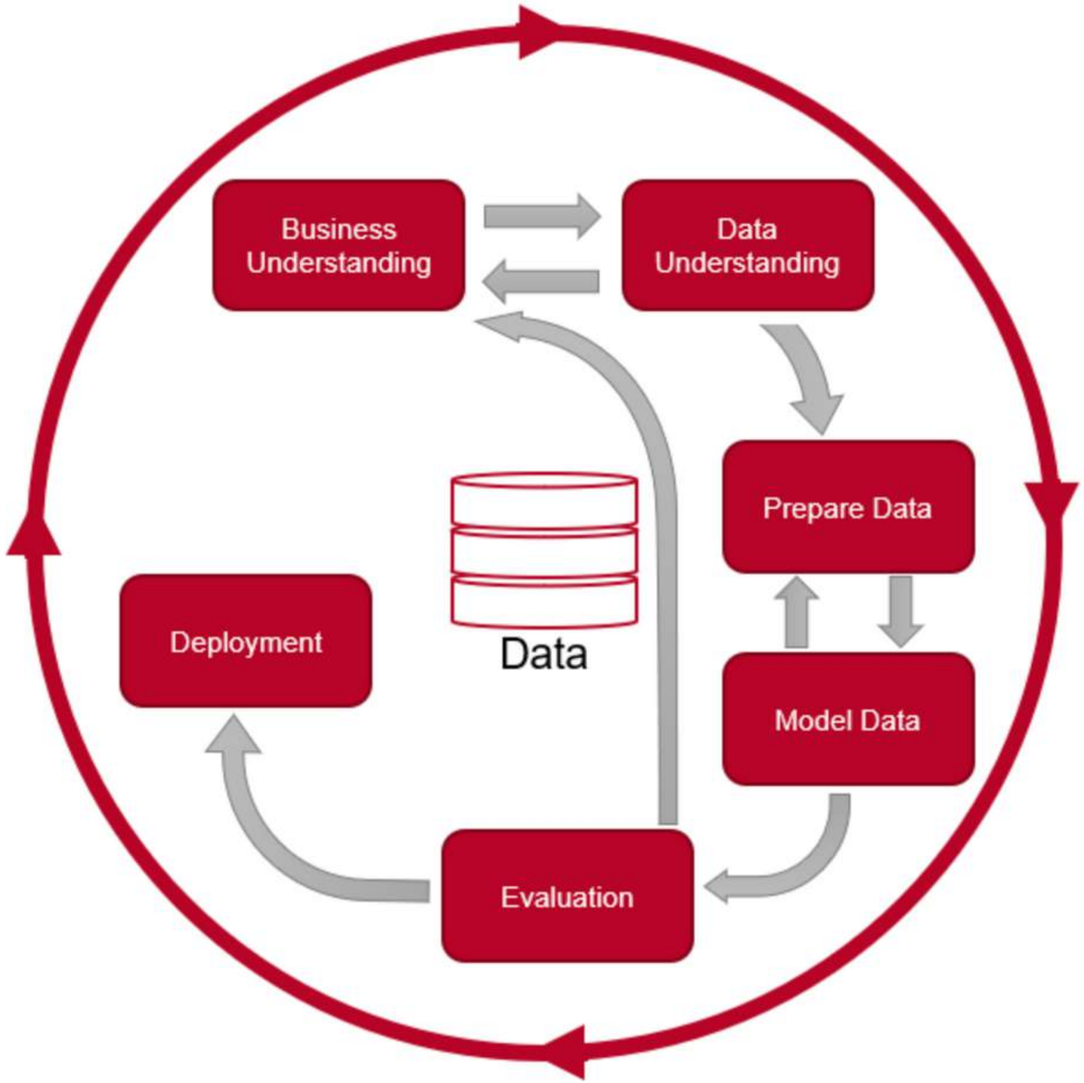
Software and Tools	Purpose	License
Jupyter Notebook	Data preprocessing, coding, and analysis	Free
Python Version 3.9	Model Development	Open Source
Libraries in Python for Machine Learning	Scikit Learn, Pandas, NumPy, XGBoost, etc.	Open Source
Zoom	Video conferencing and collaboration	Student License
Microsoft 365	Use various Microsoft tools such Microsoft Word, Excel and PowerPoint	Student License
Canva	Presentation and visual design	Free Version
Notion	Collaborative Project Scheduling and Management Tool	Free
Google Colab	Allows multiple users to collaborate and integrate in real-time.	Free
GitHub	Hosting Service to store code repositories towards the end of the Project.	Free Plan
Lucidchart/TeamGantt	Create flowcharts for CRISP-DM data flow, project management	Free version

2.5 PROJECT SCHEDULE

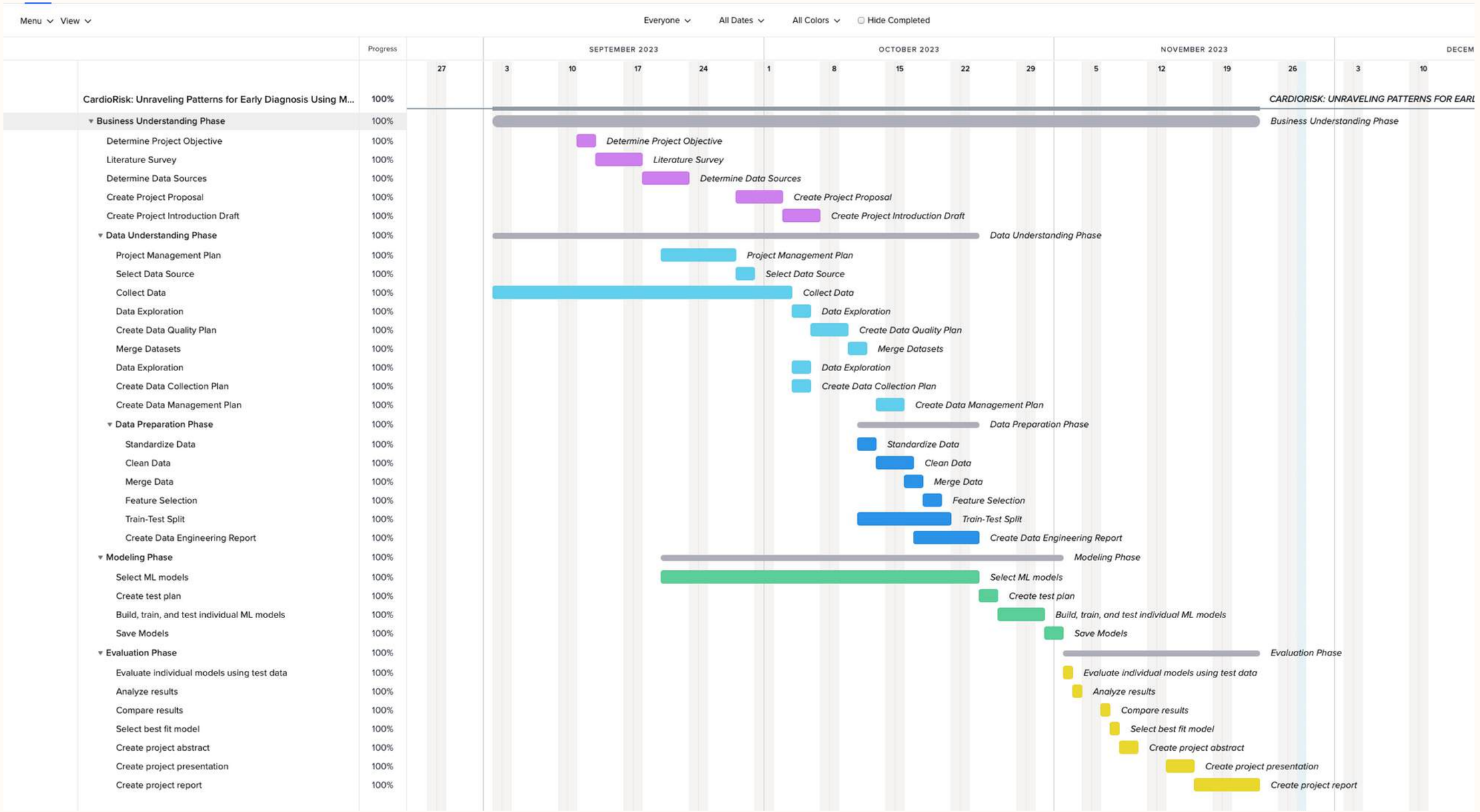
CRISP-DM METHODOLOGY

(Cross Industry Standard Process for Data Mining)

- Scheduled mapped out for 6 important phases
  - Business Understanding
  - Data Understanding
  - Data Preparation
  - Modeling
  - Evaluation
  - Deployment



2.5 PROJECT SCHEDULE USING GANTT CHART

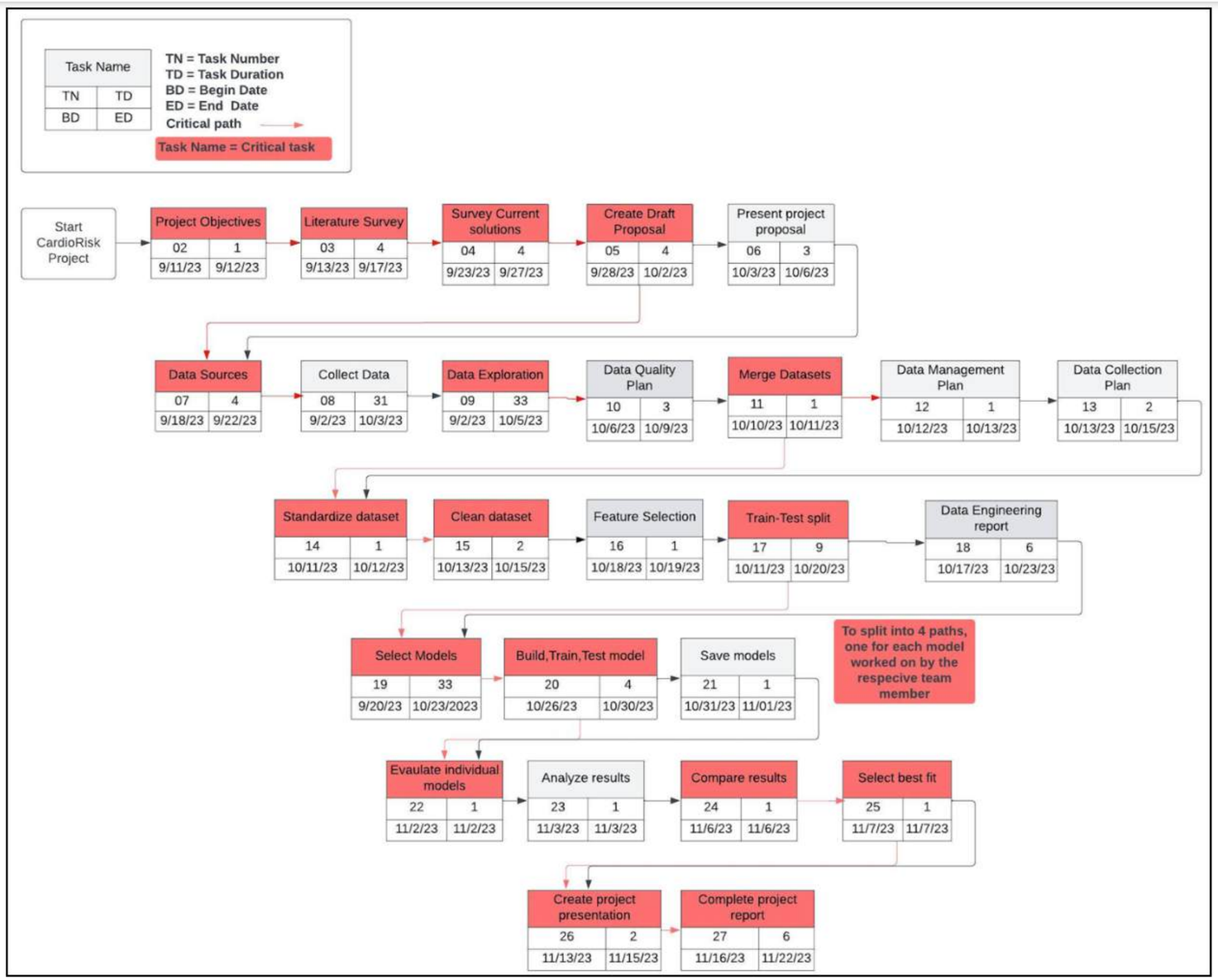


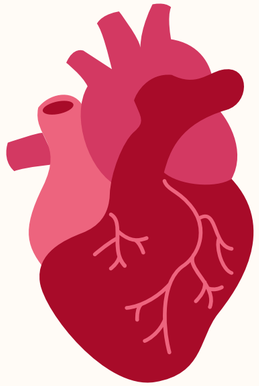
2.5 PROJECT SCHEDULE USING PERT CHART

PERT or Program Evaluation and Review Technique chart

Graphical representation of project schedule

Complete view of tasks and schedule





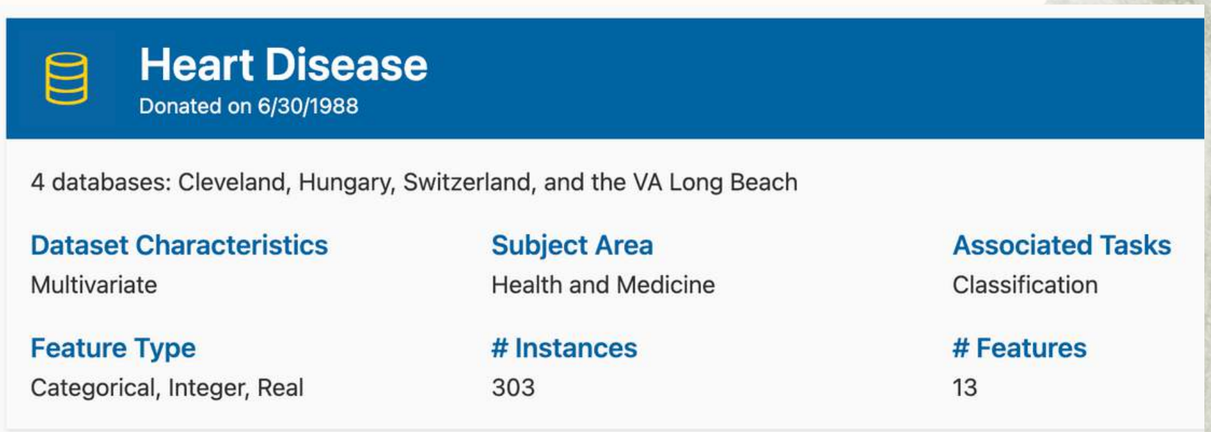
CHAPTER 3 - DATA ENGINEERING

3.1 DATA PROCESS

Source of Data	• UCI ML Repository - Heart Disease Dataset • Known for reliability and extensive use in academic studies
Data Evaluation Criteria	• Selection based on data reputation and depth of information • Preference for data linked to sources due to health-related implications
Data Collection & Preprocessing	• Conversion to CSV for better compatibility • Merging of four distinct datasets into a single dataset • Comprehensive data cleaning and pre-processing ○ Identification of null values, missing values and outliers etc.
Data Transformation & Preparation	• Applied One Hot Encoding to certain features • Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) - Baseline model evaluation • Split data - Train, Validation and Test Sets
Data Statistics	• Understanding & Visualising Final Transformed Data

3.2 DATA COLLECTION

- **76 features from 4 locations: Cleveland, Hungary, Switzerland, VA Long Beach.**
- **"Location" feature added before merging data.**
- **Raw Dataset files are in .DATA format.**



Heart Disease
Donated on 6/30/1988

4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Health and Medicine	Classification
Feature Type	**# Instances**	**# Features**
Categorical, Integer, Real	303	13

Aspect	Description
Why are we collecting data?	The UCI Heart Disease dataset is collected to facilitate the development of predictive models for heart disease. It includes a range of clinical variables critical for identifying heart disease risk factors.
How will the data help?	By applying machine learning techniques to the dataset, we aim to uncover patterns that can predict the presence or absence of heart disease, potentially aiding in early diagnosis and personalized treatment strategies.
What should we do after collecting data?	The data should be preprocessed which includes normalization, handling missing values, and categorical data encoding. The dataset must be divided into a training set for model development, a validation set for validating model performance and a testing set for model evaluation.

3.2 DATA COLLECTION

Variable title	**1 (age)**	**2 (sex)**	**3 (cp)**	**4 (trestbps)**	**5 (chol)**	**6 (fbs)**	**7 (restecg)**	**8 (num)**
Input (X) or output (Y) variable?	X	X	X	X	X	X	X	Y
Unit of measurement	Years	Binary	Chest Pain Type	mm Hg	mg/dl	Binary	Result from ECG	Diagnosis
Data type	Integer	Categorical	Categorical	Integer	Integer	Categorical	Categorical	Categorical
Collection method	**Data is downloaded from the UCI Machine Learning Repository, then processed and converted from .DATA format to .CSV.**							
Historical data exists?	**No. First documentation dates back to 1988 according to files downloaded by the team. Few updates to the database have been made since then.**							
Operational definition exists?	**Operational definition exists for the Heart Disease Dataset to ensure that data collected is consistent. Complete attribute information has been provided which are mapped towards end of data collection process. Each location's data is available in a separate file to ensure separation.**							
Data collectors	Shruti Badrinarayanan				Vani Kancherapalli			
Start date	19-Sept-2023				19-Sept-2023			
Due date	20-Sept-2023				20-Sept-2023			
Duration (in days)	1				1			

3.2 DATA COLLECTION

Sample Record from Raw Dataset (Cleveland)

```
1   1 0 63 1 -9 -9 -9
2   -9 1 145 1 233 -9 50 20
3   1 -9 1 2 2 3 81 0
4   0 0 0 0 1 10.5 6 13
5   150 60 190 90 145 85 0 0
6   2.3 3 -9 172 0 -9 -9 -9
7   -9 -9 -9 6 -9 -9 -9 2
8   16 81 0 1 1 1 -9 1
9   -9 1 -9 1 1 1 1 1
10  1 1 -9 -9 name
```

Instance Counts for Each Location

```
Location
Hungary          294
Cleveland        282
LongBeachVA      200
Switzerland      123
Name: count, dtype: int64
```

Sample Records from Dataset saved as a CSV (Switzerland)

	Location	0	1	2	3	4	5	6	7	8	...	66	67	68	69	70	71	72	73	74	75
0	Switzerland	3001	0	65	1	1	1	1	-9	4	...	1	1	1	1	1	1	1	75.0	-9.0	name
1	Switzerland	3002	0	32	1	0	0	0	-9	1	...	1	1	1	1	1	5	1	63.0	-9.0	name
2	Switzerland	3003	0	61	1	1	1	1	-9	4	...	2	1	1	1	1	1	1	67.0	-9.0	name
3	Switzerland	3004	0	50	1	1	1	1	-9	4	...	1	1	1	1	1	5	4	36.0	-9.0	name
4	Switzerland	3005	0	57	1	1	1	1	-9	4	...	2	1	1	1	1	1	1	60.0	-9.0	name

5 rows × 77 columns

Merged Dataset

	Location	id	ccf	age	sex	painloc	painexer	relrest	pncaden	cp	...	rcaprox	rcadist	lvx1	lvx2	lvx3	lvx4	lvf	cathef	junk	name
0	Cleveland	1	0	63	1	-9	-9	-9	-9	1	...	1	1	1	1	1	1	1	-9.0	-9.0	name
1	Cleveland	2	0	67	1	-9	-9	-9	-9	4	...	1	1	1	1	1	1	1	-9.0	-9.0	name
2	Cleveland	3	0	67	1	-9	-9	-9	-9	4	...	2	2	1	1	1	7	3	-9.0	-9.0	name
3	Cleveland	4	0	37	1	-9	-9	-9	-9	3	...	1	1	1	1	1	1	1	-9.0	-9.0	name
4	Cleveland	6	0	41	0	-9	-9	-9	-9	2	...	1	1	1	1	1	1	1	-9.0	-9.0	name

5 rows × 77 columns

3.3 DATA PRE-PROCESSING

- Important step in CRSIP-DM

- This stage involves cleaning and transforming raw data into suitable format for modeling or performing any other analysis

- Check for duplicate data

```
No duplicate records in dataset
```

- Check for missing values

```
Empty DataFrame
Columns: [Feature, Missing Percentage]
Index: []
```

- Check the cardinality of each feature to get more insight
- Dig deeper into features that required more analysis
- Target feature has 5 unique values

```
age 50
sex 2
painloc 3
painexer 3
relrest 3
pncaden 1
cp 4
trestbps 61
htn 3
chol 214
smoke 3
```

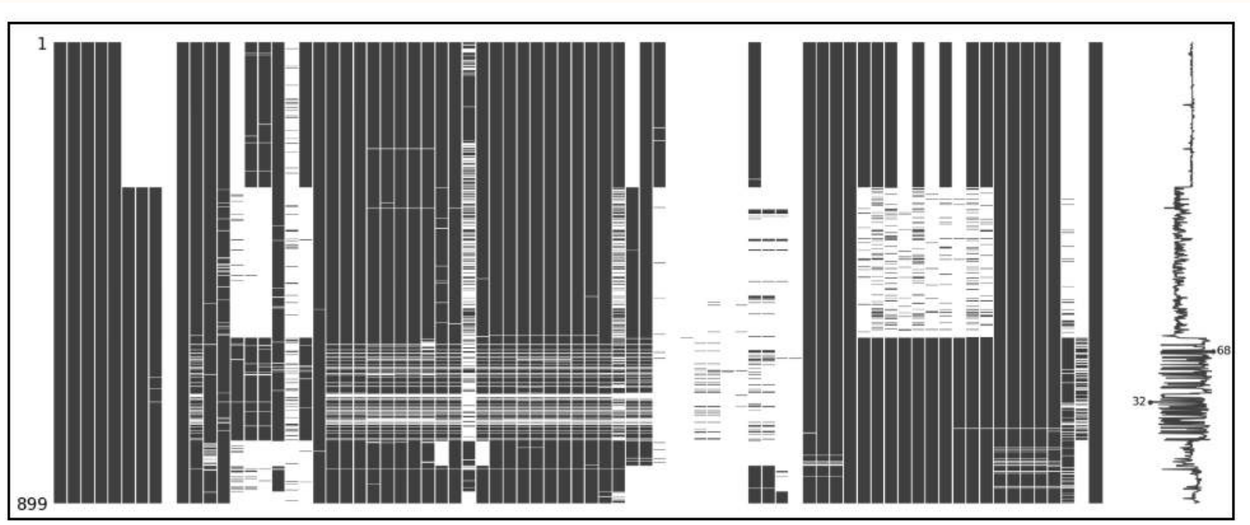
3.3 DATA PRE-PROCESSING

Special code (-9) in dataset

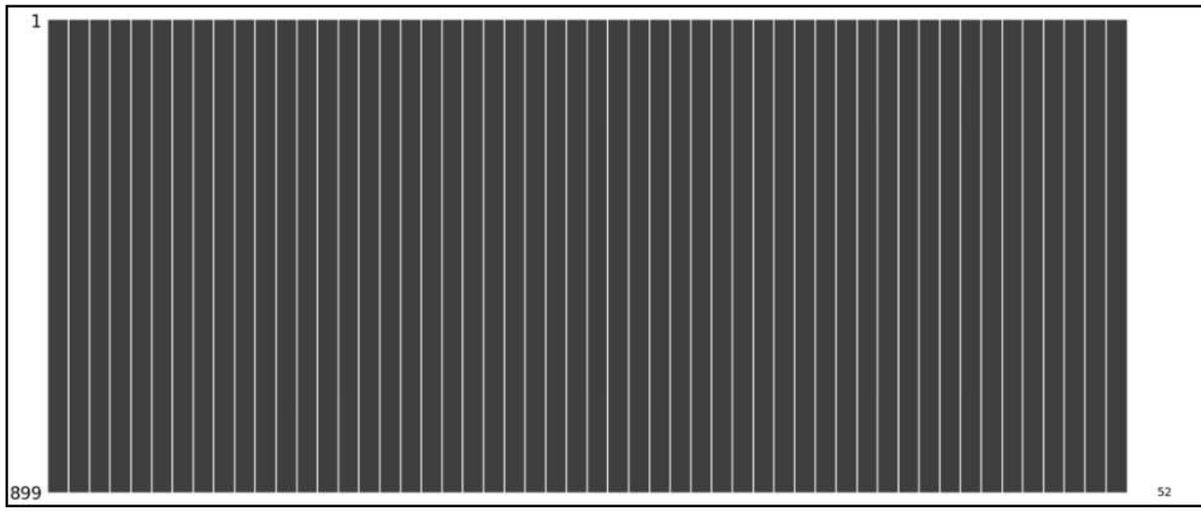
	Location	id	ccf	age	sex	painloc	painexer	relrest	pncaden	cp	...	rcaprox	rcadist	lvx1	lvx2	lvx3	lvx4	lvf	cathef	junk	name
0	Cleveland	1	0	63	1	-9	-9	-9	-9	1	...	1	1	1	1	1	1	1	-9.0	-9.0	name
1	Cleveland	2	0	67	1	-9	-9	-9	-9	4	...	1	1	1	1	1	1	1	-9.0	-9.0	name
2	Cleveland	3	0	67	1	-9	-9	-9	-9	4	...	2	2	1	1	1	7	3	-9.0	-9.0	name

missingno matrix to visualize missing data (special code -9)

Before



After



Supplementary cleaning tasks

Data imputation by mean and mode by checking cardinality	Drop unnecessary features like ‘name, ‘ccf’	Drop features missing more than 20% of values

3.3 DATA PRE-PROCESSING

Outliers in dataset documented. The models used are robust to outliers.

```
For the feature trestbps, No of Outliers is 28
For the feature chol, No of Outliers is 185
For the feature proto, No of Outliers is 2
For the feature thaldur, No of Outliers is 40
For the feature met, No of Outliers is 73
For the feature thalach, No of Outliers is 3
For the feature thalrest, No of Outliers is 13
For the feature tpeakbps, No of Outliers is 11
```

```
restecg      2
ca         608
lvf         16
dtype: int64
```

The IterativeImputer from scikit-learn library is used when dealing with datasets that contain missing values and when you want to impute those missing values based on the observed values in those columns

3.4 DATA TRANSFORMATION

One Hot Encoding

Encoding categorial feature

	Location_Cleveland	Location_Hungary	Location_LongBeachVA
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0

Encode new target feature ‘num_encoded’

```
0    404
1    191
3    132
2    130
4     42
Name: num, dtype: int64
```

```
1    495
0    404
Name: num_encoded, dtype: int64
```

Dimensionality Reduction

- Use of PCA and LDA
- PCA - Crucial for small datasets - Prevents Overfitting
- LDA - Enhances Class Separation
- Samples of Training Dataset after standardisation :

	age	sex	painloc	painexer	relrest	cp	trestbps	htn	chol	fbs	...	cxmain	om1	rcaprox	rcadist	lvx1	lvx2	lvx3	lvx4	lvf	target
0	2.150388	0.511276	-4.420730	-1.627804	-1.876703	-1.391787	0.676765	1.067652	-1.812612	-0.431820	...	-0.547888	-0.383054	-0.580027	-0.375859	0.030945	0.000538	0.055681	0.003910	1.687619	1.0
1	0.892606	0.511276	0.226207	-1.627804	-1.876703	-1.391787	0.409648	-0.936635	0.647353	-0.431820	...	-0.547888	-0.383054	-0.580027	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	0.0
2	1.102236	0.511276	0.226207	0.614325	0.532849	0.802720	-0.124587	1.067652	0.529347	2.315779	...	-0.547888	-0.383054	1.724058	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	1.0
3	-0.365177	0.511276	0.226207	0.614325	0.532849	0.802720	0.409648	1.067652	-0.641633	-0.431820	...	-0.547888	-0.383054	-0.580027	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	0.0
4	0.997421	0.511276	0.226207	0.614325	0.532849	0.802720	-0.124587	-0.936635	1.182917	2.315779	...	1.825192	2.610597	1.724058	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
714	0.578160	0.511276	0.226207	0.614325	0.532849	0.802720	0.409648	1.067652	-0.205919	-0.431820	...	1.825192	-0.383054	-0.580027	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	1.0
715	0.787791	0.511276	0.226207	0.614325	0.532849	0.802720	0.409648	-0.936635	0.066401	-0.431820	...	-0.547888	-0.383054	-0.580027	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	1.0
716	1.207051	0.511276	0.226207	0.614325	0.532849	0.802720	1.478117	1.067652	-1.812612	2.315779	...	-0.547888	-0.383054	1.724058	-0.375859	-0.072970	-0.075728	-0.171137	3.167493	1.687619	1.0
717	0.158899	0.511276	0.226207	0.614325	0.532849	0.802720	-0.658822	-0.936635	0.638275	-0.431820	...	-0.547888	-0.383054	-0.580027	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	0.0
718	0.368530	-1.955889	0.226207	0.614325	0.532849	0.802720	-0.231434	1.067652	0.937828	-0.431820	...	-0.547888	-0.383054	-0.580027	-0.375859	-0.072970	-0.075728	-0.171137	-0.355012	-0.348856	0.0

719 rows × 51 columns

3.4 DATA TRANSFORMATION

Dimensionality Reduction using PCA

Overall Comparison

- **Baseline Model : Random Forest**
- **Decision: Chose PCA for its effectiveness in the given experiments.**
- **Runtime: Increased with component count, indicating a trade-off.**
- **Final Component Count : 40**

Type	Number of Components	Accuracy	Run-TIme in Seconds
PCA	5	0.905	0.168
PCA	25	0.922	0.266
PCA	30	0.905	0.270
PCA	35	0.927	0.310
PCA	40	0.933	0.307
PCA	45	0.911	0.309
PCA	50	0.933	0.368

3.5 DATA PREPARATION

Sample from the Training Dataset

	1	2	3	4	5	6	7	8	9	10	...	32	33	34	35	36	37	38	39	40	target
0	-0.766935	-2.353474	0.666855	1.790006	2.471166	2.477473	0.962876	-2.439171	0.404615	-1.986088	...	1.710136	0.313500	-0.651644	-0.216032	-0.820337	-0.844128	-0.591588	0.538382	0.012462	1.0
1	-3.732850	-2.177280	-1.935404	1.336965	-0.652893	0.839545	0.218358	-0.386887	-0.144375	-0.500492	...	-0.443013	0.311716	-0.672536	-1.066886	-0.055299	0.003194	-0.539007	0.788195	-0.187139	0.0
2	1.843476	-0.389318	-0.046146	1.063272	0.402922	-0.240648	0.047470	0.841115	0.135274	-2.172294	...	-0.168109	-0.205389	-0.774231	-0.651174	0.174469	-0.121232	0.297091	-1.092843	-0.510920	1.0
3	-2.048637	-0.149163	0.158482	-1.459777	-1.914820	-0.667933	-0.366468	1.192296	1.081936	-0.271876	...	0.575804	-0.208756	0.483045	0.270546	0.331686	-0.144162	0.006243	0.144280	-0.086948	0.0
4	4.328346	1.455783	-1.130231	0.692913	-0.472766	1.465060	-0.779669	-2.449843	-1.778051	-0.790932	...	-0.673324	0.403204	-0.510762	1.919140	-0.062170	-0.202954	-0.479562	0.876458	0.032876	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
714	1.762246	1.532235	0.036873	0.364361	-1.307962	0.481195	0.663398	0.225311	-0.153818	-0.526096	...	-0.810439	0.109616	-1.039656	-0.139942	1.744363	0.886578	0.243788	-0.111357	-0.340558	1.0
715	1.287805	1.787164	-0.403692	0.910878	-1.444662	0.448444	-0.452303	-1.021009	-0.918742	0.508371	...	-0.725539	-0.616482	-1.587135	0.294064	-0.221532	-0.403016	0.156068	-0.448974	-0.259074	1.0
716	4.576414	-1.701957	0.571544	2.505701	1.047498	2.503107	0.723274	2.139162	1.321981	-4.264630	...	0.634040	-1.640712	-0.407667	-1.701723	-1.475635	-0.548300	1.359511	0.597624	0.928807	1.0
717	-2.055850	-0.752798	0.085880	-2.090918	-0.781065	-1.193076	2.593513	0.790864	-0.904424	-0.437684	...	0.464635	0.168204	0.303898	-0.379718	0.230896	0.161814	0.237173	0.280401	-0.086077	0.0
718	-1.154553	3.183363	-0.157419	0.639956	-0.572280	0.132195	-0.450627	0.626799	0.271958	0.168394	...	-0.071319	0.088282	-0.856900	-0.189416	0.260927	-0.030486	-0.175321	0.424356	-0.103499	0.0

719 rows × 41 columns

Sample from the Validation Dataset

	1	2	3	4	5	6	7	8	9	10	...	32	33	34	35	36	37	38	39	40	target
0	4.027235	-1.208855	2.382666	-1.490612	1.471426	1.946134	-0.463684	0.575949	0.687090	2.246827	...	1.010056	-3.212606	0.060761	-4.535548	-0.708055	-0.212160	-0.110899	0.471038	1.612919	1.0
1	-0.299588	2.740717	-0.883797	-1.051960	0.612017	-0.261867	0.589333	0.830389	2.029004	-1.118576	...	0.403736	-0.411271	1.175541	0.199788	-0.709673	-0.057885	0.273499	0.290625	0.671585	0.0
2	4.742652	-2.150197	2.022480	2.053729	-1.942904	0.421636	0.950476	-0.530450	0.746796	-0.592331	...	-0.116253	-1.028769	-0.284396	-0.248165	-0.200482	0.092726	-0.065671	-0.440314	-0.387434	1.0
3	3.642856	-1.335302	2.745068	0.348092	-4.206645	3.516037	-0.433059	-2.082450	0.098109	-0.741439	...	-0.686204	-0.684754	-1.319149	0.625995	-1.768072	1.458167	-0.330908	0.066761	1.491952	1.0
4	9.068448	-4.371941	3.567319	-4.876332	4.186393	10.287135	-2.649089	7.947423	-3.699511	4.207660	...	-1.030756	-3.080694	-0.421101	-1.159176	-0.667355	1.178084	-1.750864	1.015000	-0.588737	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
85	2.612977	-2.045695	0.276530	-0.570358	2.233862	0.482197	-0.250445	3.691157	-3.380202	-1.572645	...	-0.600315	-0.744156	-1.818869	-0.265212	-1.565192	0.437648	-2.591140	1.840803	-1.519305	1.0
86	0.763549	0.830619	1.625899	-2.102867	0.570731	-2.224988	1.020102	-0.245991	-0.562334	0.660858	...	-0.810355	1.144991	0.500716	0.500708	-0.109506	-0.509059	-0.514900	-0.339596	-0.361709	1.0
87	-3.682426	-1.213354	1.599450	-0.401168	1.170228	-0.412823	0.760563	-0.779524	0.075499	0.455980	...	-0.424761	0.367584	-1.181349	-0.390730	0.085740	0.066691	0.023064	0.208902	0.031795	0.0
88	1.129205	1.025513	2.163184	-0.073920	-0.998707	-1.316303	1.415864	0.662023	1.577756	-0.930364	...	-1.043207	0.606110	-1.399164	-0.334714	0.632390	-1.374277	-0.452751	0.847350	0.081500	0.0
89	0.927789	0.148099	-0.666059	-0.070758	-0.168624	-0.323310	-0.121696	-0.577819	-0.382220	-0.795173	...	0.790104	-0.704514	-0.172941	0.930994	0.709396	0.395248	-0.562216	0.065953	0.960717	1.0

90 rows × 41 columns

Sample from the Test Dataset

	1	2	3	4	5	6	7	8	9	10	...	32	33	34	35	36	37	38	39	40	target
0	1.963897	3.312018	-0.492742	0.716432	0.500319	-0.667362	1.842020	-0.149831	0.149125	0.247344	...	-0.376934	-0.739054	-1.299408	-0.078289	0.736444	0.574571	0.391968	-1.113413	0.192045	1.0
1	2.605402	-0.010679	-0.465319	1.484976	-0.161494	0.168188	1.773480	1.050517	2.742299	-2.438317	...	0.199789	-0.341554	-0.202215	0.495685	1.251404	-0.912478	-0.012528	-0.887978	-0.594072	1.0
2	1.465118	1.901726	0.924220	-2.026664	-1.002603	0.007550	-0.973233	-1.583329	-1.028633	-0.515163	...	0.362198	-0.453499	-0.029237	-0.556419	0.769007	-0.404074	-0.062946	-1.099522	-0.334097	1.0
3	-3.942047	-0.208087	1.525755	-1.393688	1.369831	-0.149219	0.091647	-0.487857	-0.063387	0.208683	...	-0.405234	-0.392000	0.846692	0.212451	0.023352	0.284554	0.005036	-0.413273	0.018656	0.0
4	2.444208	0.407474	-3.205163	1.137219	-1.037876	1.046500	-1.041223	-0.712243	-0.040577	1.055758	...	-0.769215	-0.176596	-0.803753	-0.195633	-0.112768	-0.041968	-0.675655	-0.541130	0.583435	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
85	-0.860501	1.944474	-2.174360	1.097662	0.281725	1.619351	-0.912577	0.078777	2.175257	-0.870730	...	-0.079762	0.080890	0.473737	0.813514	-0.424143	1.077358	0.353909	-0.179338	0.277537	0.0
86	-4.096910	-2.866549	-1.681153	1.511321	-0.726947	0.848420	0.152375	-0.126661	-0.408773	-0.500199	...	-0.591350	-0.127535	-0.464372	-0.456002	-0.401066	0.232518	-0.145193	-0.404850	0.315576	0.0
87	-1.871070	2.930604	-0.306529	-0.233074	-0.223521	0.677403	-0.300307	0.009680	-0.049187	-0.220039	...	-0.191732	-0.293560	0.662368	0.031777	-0.124635	0.452857	0.426775	-0.477175	-0.022281	0.0
88	-1.882366	2.864284	-0.729686	2.321626	0.458405	0.292697	0.186423	0.556544	-1.724291	-0.261264	...	-0.168151	0.604138	0.189979	-0.849006	-1.523453	-0.442773	-0.593973	-0.345359	0.978618	0.0
89	2.732532	1.710480	-1.079449	-0.030737	-0.210170	1.094822	-0.126636	-1.412591	-0.977046	-0.461310	...	-0.056780	0.730190	-0.477042	0.105322	-1.208439	-0.396080	-0.587205	0.208510	0.332619	1.0

90 rows × 41 columns

3.6 DATA STATISTICS

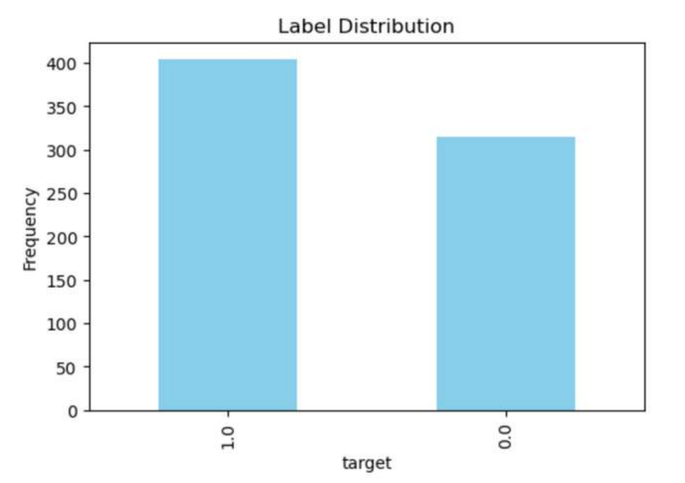
Summary of Dataset Sizes After Different Processes

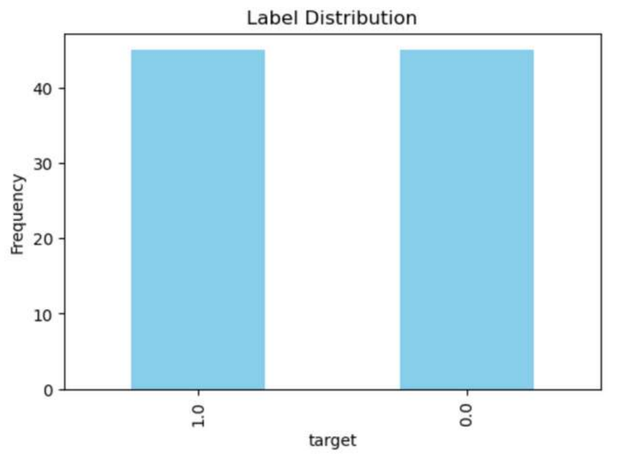
- **Final Dimensions of the data before and after each stage.**

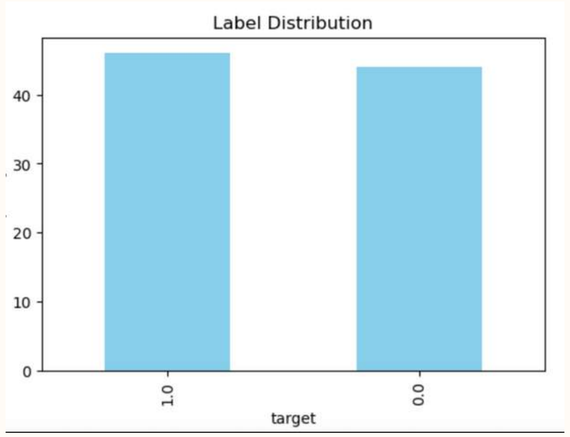
Stage	Process	After process [Row X Col]
Raw	Cleveland	282 X 75
	Hungary	294 X 75
	Switzerland	123 X 75
	Long Beach VA	200 X 75
Pre-processing	Cleveland	282 X 52
	Hungary	294 X 52
	Switzerland	123 X 52
	Long Beach VA	200 X 52
	Merge	899 X 52
Transformation	Dimensionality Reduction using PCA	41
Preparation	Training Data	719 X 41
	Testing Data	90 X 41
	Validation Data	90 X 41

3.6 DATA STATISTICS

- **Data statistics shows the results of the transformed dataset.**
- **Split dataset 80:10:10**
  - **Train set, Rows : 719 and Column : 41**
  - **Test set, Rows : 90 and Column : 41**
  - **Val set, Rows : 90 and Column : 41**

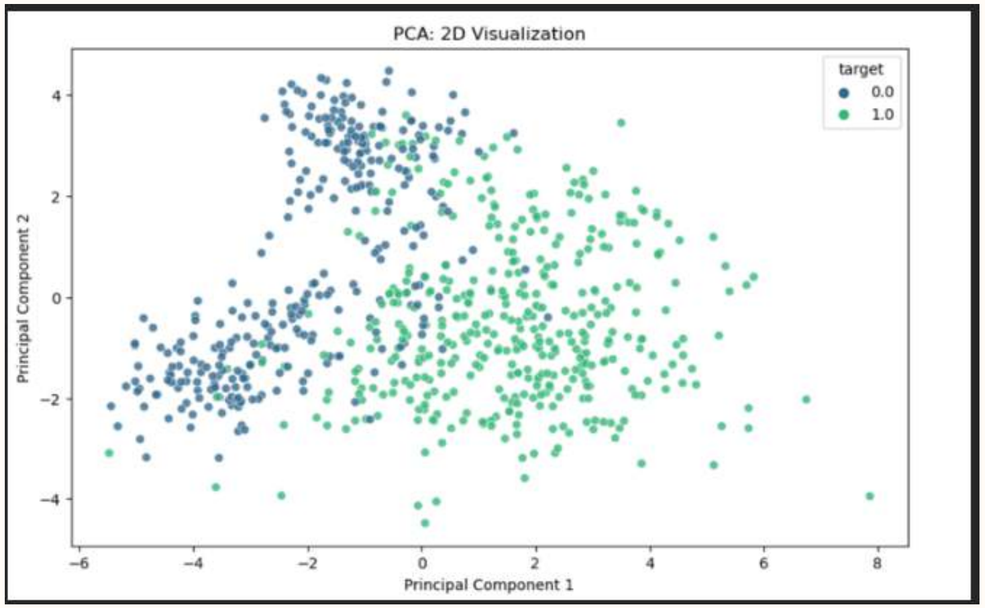


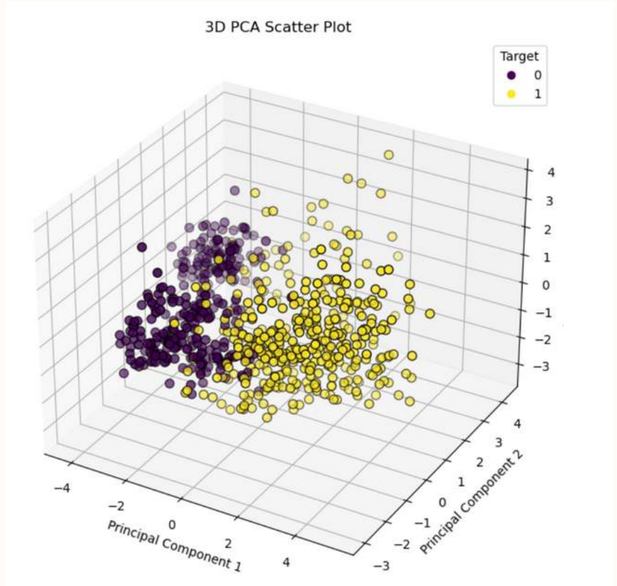




3.6 DATA STATISTICS

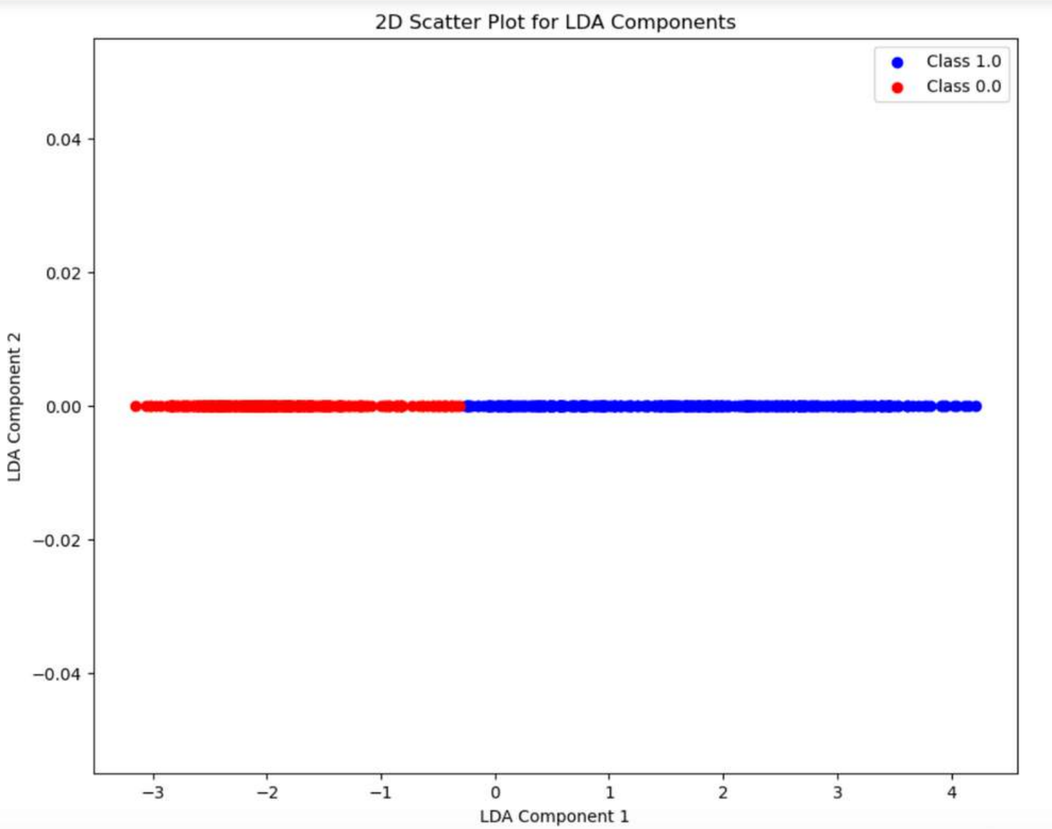
- **Scatter plots: Visualising PCA results in 2D and 3D respectively :**

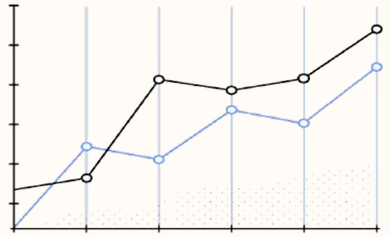




3.6 DATA STATISTICS

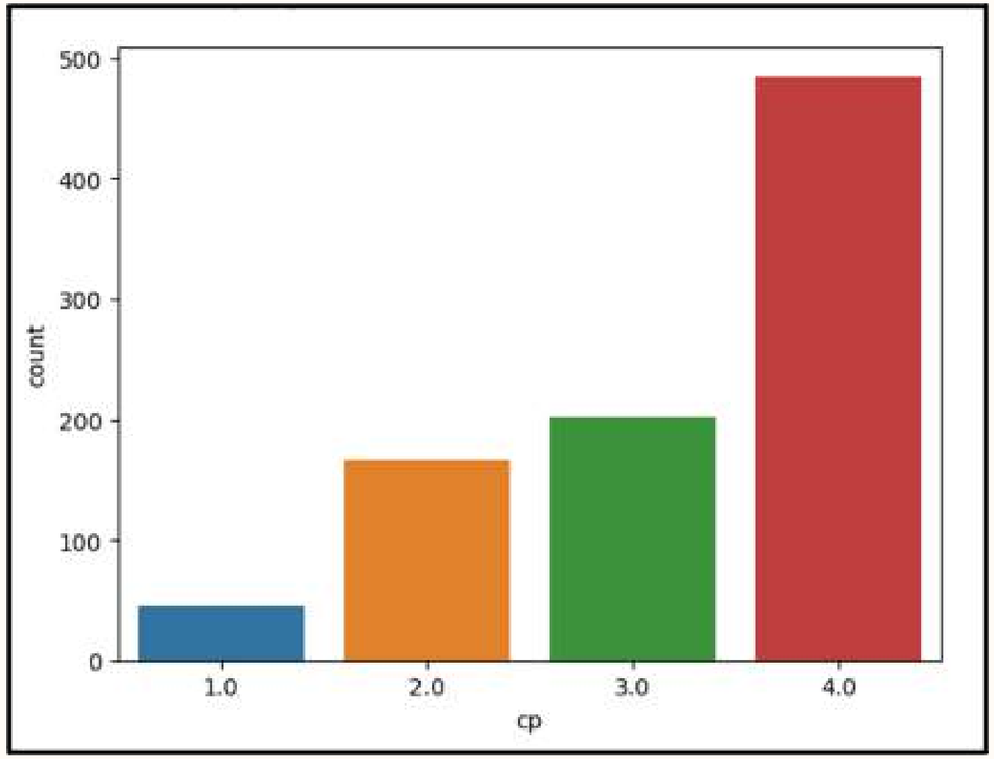
- **Discriminative performance of class 0 and 1 after LDA.**

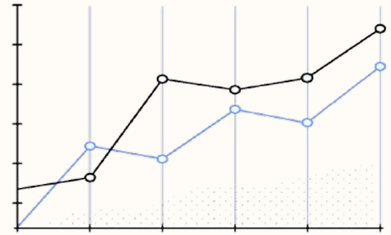


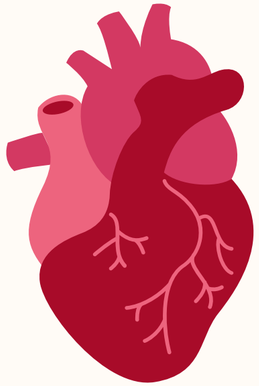


3.6 DATA STATISTICS

- Bar graph for different types of chest pain.
- 1 (typical angina), 2 (atypical angina), 3 (non-anginal pain) and 4 (asymptomatic)







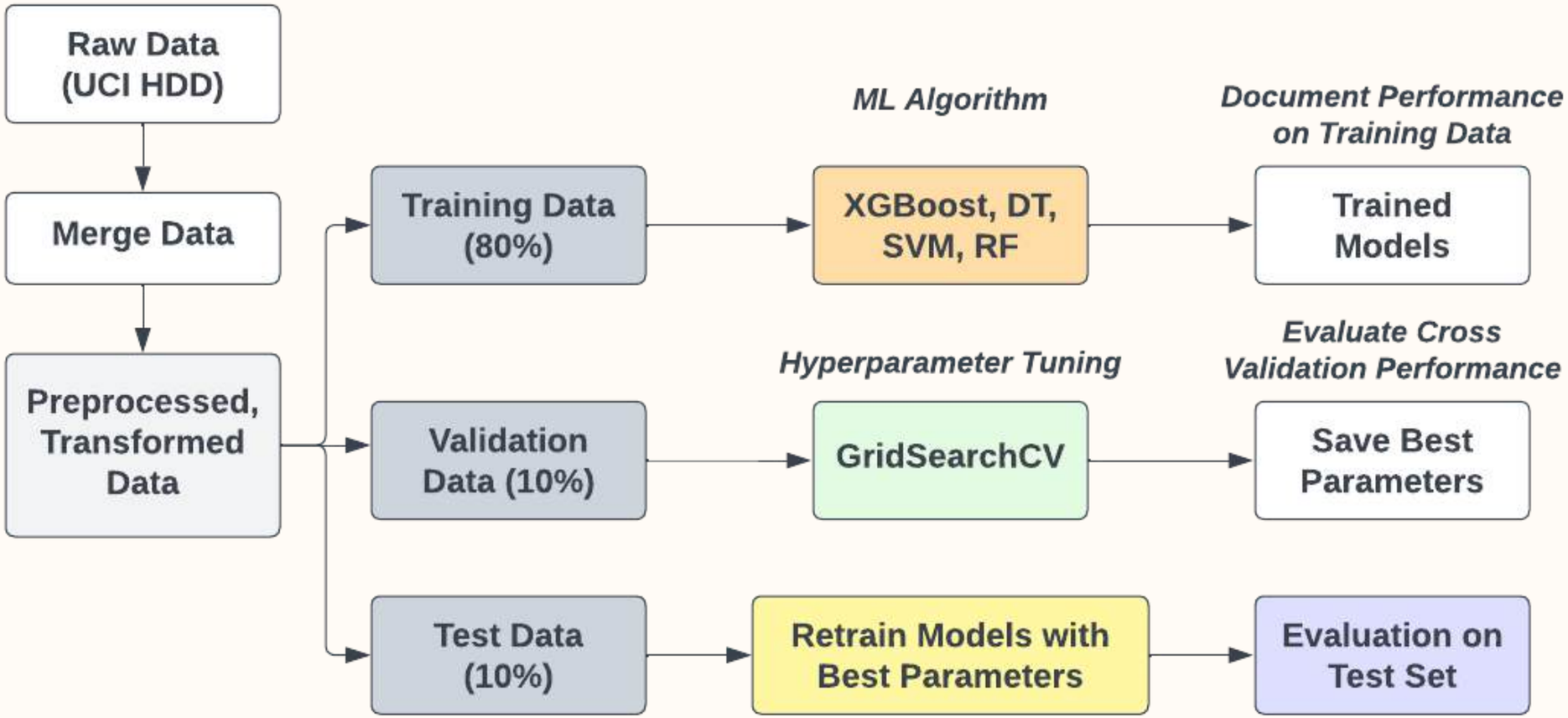
CHAPTER 4 - MODEL DEVELOPMENT

4.1 MODEL PROPOSALS

Model	Key Characteristics	Hyperparameters	Remarks on Pseudocode/Implementation
XGBoost	• Gradient boosting framework • Handles missing data • Includes regularization	• max_depth • learning_rate • n_estimators • subsample	• Iterative boosting of decision trees • Regularization terms in objective function
Decision Trees	• Simple and interpretable • Non-parametric model	• max_depth • min_samples_split • criterion	• Greedy approach to split nodes • No iterative process; single tree built
SVM	• Effective in Handling Non-Linearity. • High Dimensional Data • Cross-Validation	• C (regularization parameter) • Kernel- rbf	• Optimization of a convex function • Support vectors identified for margin maximization
Random Forest	• Ensemble of decision trees • Reduces overfitting • Good for handling imbalanced datasets	• n_estimators (RF) • max_features • min_samples_split • min_samples_leaf • bootstrap	• Builds multiple decision trees and combines their predictions • Each tree is built on a random subset of data and features • Majority voting or averaging for final prediction

4.2 MODEL SUPPORTS

System Architecture and Model Support Diagram



4.2 ENVIRONMENT, PLATFORM, AND TOOLS

Purpose	Libraries Used	Description
Data Manipulation	Pandas, Numpy, DataFrame	They offer efficient data manipulation tools for analysis and data manipulation.
Data Visualization	matplotlib, seaborn	Plotting various visulaizations
Data Preprocessing	scikit-learn	To effectively prepare and explore datasets before model training.
Model Building and Training	train_test_split	Split the training data for training and validation
Model Evaluation	from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, roc_curve, roc_auc_score, mean_squared_error,log_loss, precision_score, recall_score, f1_score	A comprehensive set of tools for assessing the performance of models.
Hyperparameter Tuning	scikit-learn, GridSearchCV, RandomizedSearchCV	For hyperparameter tuning, enabling systematic exploration and optimization of model parameters to enhance model performance.
Model Interpretation	SHAP, Lime	Provides insights into the nature of the models by explaining individual predictions and feature importance.

4.3 MODEL COMPARISON AND JUSTIFICATION

Model	Advantages	Disadvantages
Decision Trees	• Simple to understand and interpret. The trees can be visualized • No assumptions about data, the data distribution • Handles non-linear relationships	• Decision trees learners can create complex trees which are hard to generalize, this is called as overfitting • Instability implying small changes in data can lead to different trees • Decision trees may not capture complex relationships as well as other algorithms
Random Forests	• Ensemble learning reduces overfitting • The ensemble nature of Random Forests makes them robust to outliers in the data • Random Forests provide high accuracy for classification and regression problems	• Less interpretable compared to a single decision tree • Computational complexity during training, leading to longer training times • Imbalanced classes in classification problems, Random Forests may be biased toward the dominant class

4.3 MODEL COMPARISON AND JUSTIFICATION

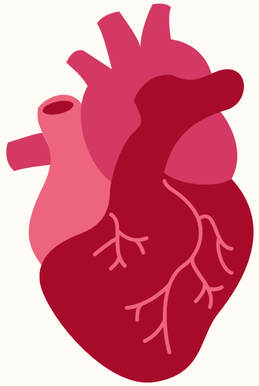
SVM	• High Performance: Excels in processing complex data, crucial for medical diagnostics • Automatic Feature Handling: Manages missing values and diverse data types effectively • Regularization: Includes features to combat overfitting in small datasets • Customization: Supports specific clinical objectives for targeted results • Model Interpretability: Offers insights into feature contributions using SHAP values	• Sensitivity to Parameter Tuning: SVMs' performance is susceptible to the choice of hyperparameters • Resource Intensive: Tuning numerous hyperparameters requires significant computing resources • Small Data Challenges: Risk Overfitting • Limited Generalization: Success on specific datasets might not extend to different datasets
XGBoost	• Excels in processing complex data, crucial for medical diagnostics • Manages missing values and diverse data types effectively • Regularisation to combat overfitting in small datasets • Model Interpretability: Offers insights into feature contributions using SHAP values	• It can overfit if regularization isn't managed properly • Tuning numerous hyperparameters requires significant compute power • Limited size of dataset raises overfitting risks • Interpretability-Accuracy Balance: While XGBoost can be made interpretable, the most accurate models may use complex ensembles of trees that are harder to interpret than simpler models

4.3 MODEL COMPARISON AND JUSTIFICATION

Model/Factors	Flexibility	Interpretability	Handling of Outliers	Parallel processing	Computational/memory intensive	Accuracy on test
Decision Tree	**Models complex non-linear**	**Easy to interpret and visualize**	**Robust**	**Limited parallelism**	**Low (storing tree)**	**83%**
SVM	**Effective for high-dimensional**	**Less intutive than decision tree**	**Sensitive**	**Limited parallelism for training**	**Low to medium (complexity of model)**	**93.33%**
Random Forest	**High flexibility, captures complex relationship**	**Less intuitive than individual tree**	**Robust**	**High parallelism**	**Moderate to high (depends on depth)**	**94%**
XGBoost	**High flexibility, captures**	**Handles, missing values, and**	**Robust**	**High parallelism**	**Medium to high (model size and**	**86.67%**

4.4 MODEL EVALUATION METHODS

Model Evaluation Metric	Key Characteristics
Confusion Matrix	• Describes the performance of a classification model. • Shows the number of true positives, true negatives, false positives, and false negatives.
Accuracy Score	• The ratio of correctly predicted instances to the total instances. • It provides an overall measure of the model's correctness. • Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision	• The ratio of true positives to the total predicted positives. • It indicates the accuracy of positive predictions. • Precision = TP / (TP + FP))
Recall	• The ratio of true positives to the total actual positives. • It measures the ability of the model to capture all relevant instances. • Recall = TP / (TP + FN)
F1- Score	• The harmonic mean of precision and recall. • Balance between Precision and Recall. • F1 Score = 2 * (Precision * Recall) / (Precision + Recall))
AUC-ROC Curve	• Quantifies the model's ability to distinguish between classes. • A higher AUC indicates better performance.

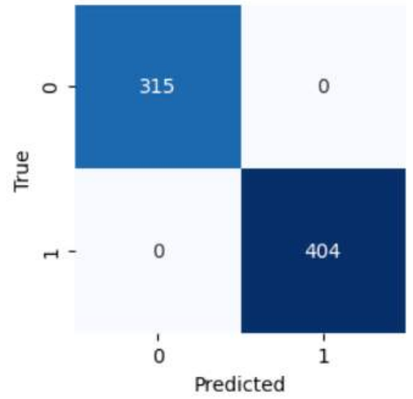


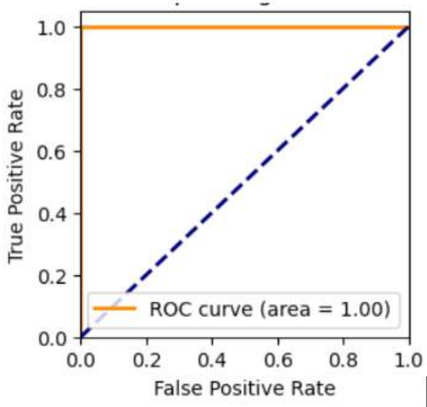
4.5 MODEL VALIDATION AND EVALUATION METHODS

XGBOOST

METRICS FOR XGBOOST BASELINE MODEL

```
Training Accuracy: 1.0000
Training Balanced Accuracy: 1.0000
Training Precision: 1.0000
Training Recall: 1.0000
Training F1 Score: 1.0000
Training ROC AUC: 1.0000
Training Log Loss: 0.0045

Training Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00       315
         1.0       1.00      1.00      1.00       404

    accuracy                           1.00       719
   macro avg       1.00      1.00      1.00       719
weighted avg       1.00      1.00      1.00       719
```



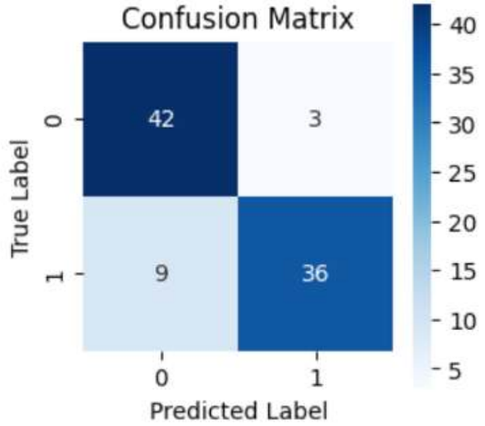


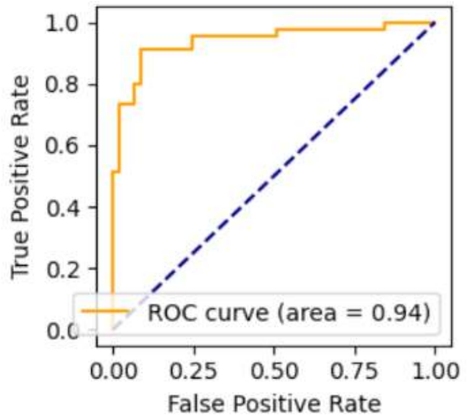
BEST PARAMETERS, VALIDATION ACCURACY OF XGBOOST HYPERPARAMETER TUNED MODEL

```
Best Parameters: {'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100, 'subsample': 1}
Best Cross Validation Accuracy: 0.9000
```

METRICS FOR XGBOOST HYPERPARAMETER TUNED WITH TEST DATA

```
Test Accuracy: 0.8667
Test Balanced Accuracy: 0.8667
Test Precision: 0.9231
Test Recall: 0.8000
Test F1 Score: 0.8571
Test ROC AUC: 0.9398
Test Log Loss: 0.3353

Test Classification Report:
              precision    recall  f1-score   support

         0.0       0.82      0.93      0.87        45
         1.0       0.92      0.80      0.86        45

    accuracy                           0.87        90
   macro avg       0.87      0.87      0.87        90
weighted avg       0.87      0.87      0.87        90
```



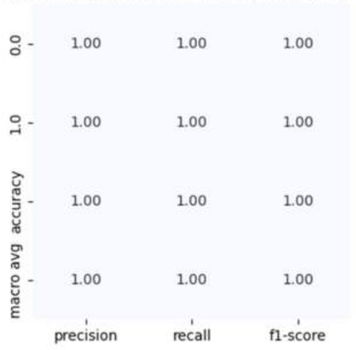


DECISION TREE

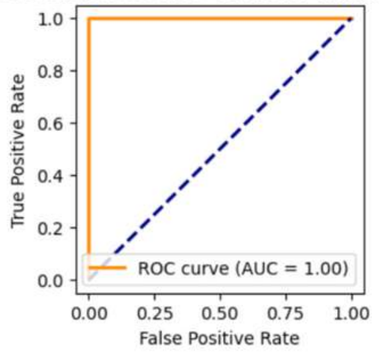
METRICS FOR DECISION TREE BASELINE MODEL

```
Decision Tree : Training Accuracy: 1.0000
Classification Report on Training Set:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00       315
         1.0       1.00      1.00      1.00       404

    accuracy                           1.00       719
   macro avg       1.00      1.00      1.00       719
weighted avg       1.00      1.00      1.00       719
```

Decision Tree : Classification Report Heatmap for Training Set



Decision Tree : ROC Curve for Training set



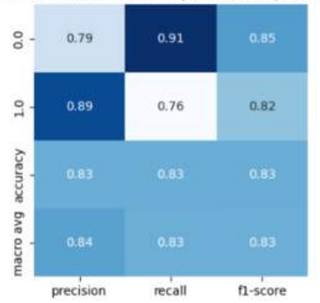
BEST PARAMETERS, VALIDATION ACCURACY OF DECISION TREE HYPERPARAMETER TUNED MODEL

```
Decision Tree Best Parameters: {'criterion': 'entropy', 'max_depth': 2, 'min_samples_leaf': 4, 'min_samples_split':
2}
Decision Tree Best Validation Accuracy: 0.7778
Decision Tree Validation Accuracy: 0.8778

Decision Tree Validation Classification Report:
              precision    recall  f1-score   support

         0.0       0.84      0.93      0.88        44
         1.0       0.93      0.83      0.87        46

    accuracy                           0.88        90
   macro avg       0.88      0.88      0.88        90
weighted avg       0.88      0.88      0.88        90
```

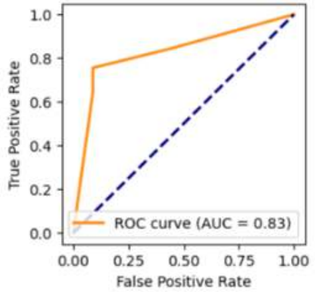
METRICS FOR DECISION TREE HYPERPARAMETER TUNED WITH TEST DATA

```
Decision Tree : Test Accuracy: 0.8333
Decision Tree : Classification Report on Testing Set:
              precision    recall  f1-score   support

         0.0       0.79      0.91      0.85        45
         1.0       0.89      0.76      0.82        45

    accuracy                           0.83        90
   macro avg       0.84      0.83      0.83        90
weighted avg       0.84      0.83      0.83        90
```

Decision Tree : Classification Report Heatmap for Testing Set



Decision Tree : ROC Curve for Testing set



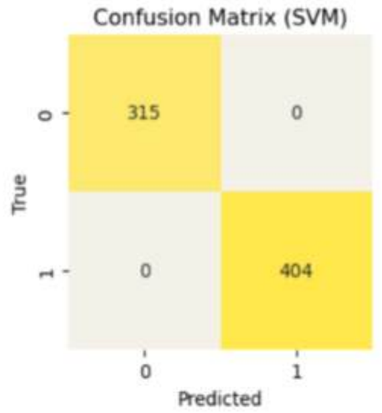
SVM

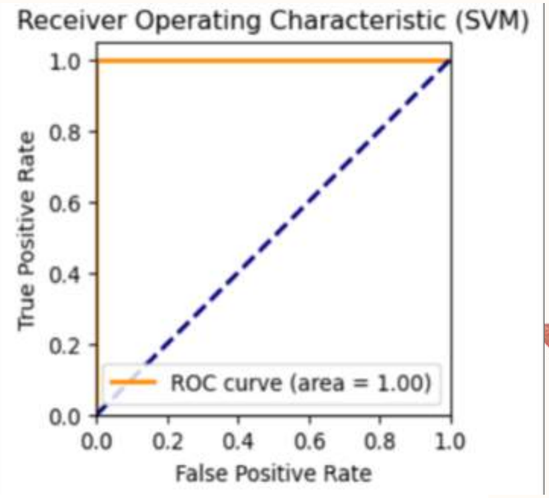
METRICS FOR SVM BASELINE MODEL

fice on the web Frame

```
Training Balanced Accuracy (SVM): 1.0000
Training Precision (SVM): 1.0000
Training Recall (SVM): 1.0000
Training F1 Score (SVM): 1.0000
Training ROC AUC (SVM): 1.0000
Training Log Loss (SVM): 0.0045

Training Classification Report (SVM):
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00       315
         1.0       1.00      1.00      1.00       404

    accuracy                           1.00       719
   macro avg       1.00      1.00      1.00       719
weighted avg       1.00      1.00      1.00       719
```

```
Training Confusion Matrix (SVM):
[[315   0]
 [  0 404]]
```





BEST PARAMETERS, VALIDATION ACCURACY OF SVM HYPERPARAMETER TUNED MODEL

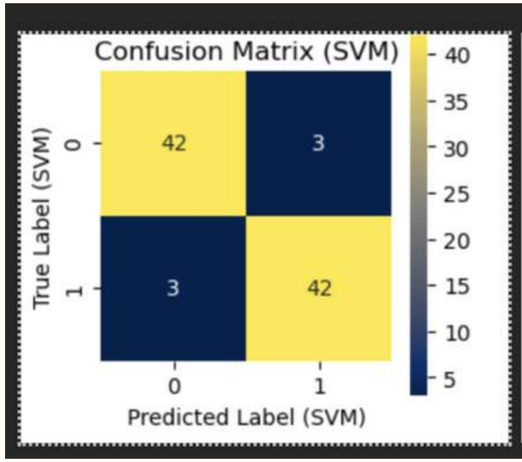
```
Best Parameters (Grid Search - SVM): {'C': 10, 'class_weight': None, 'gamma': 'scale', 'kernel': 'rbf'}
Best Cross Validation Accuracy (Grid Search - SVM): 0.9889
Validation Accuracy (SVM): 0.9778
```

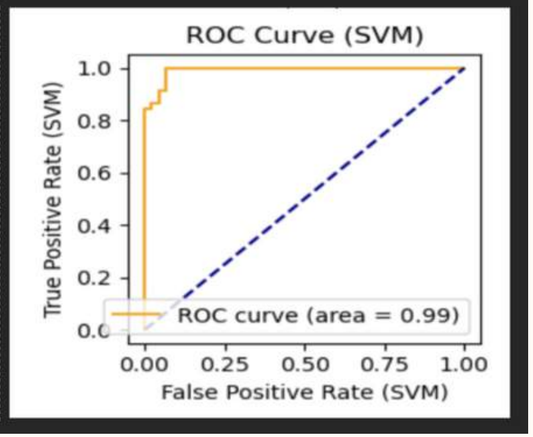
METRICS FOR SVM HYPERPARAMETER TUNED WITH TEST DATA

Metrics for SVM Model with Test Data

```
Test Accuracy (SVM): 0.9333
Test Balanced Accuracy (SVM): 0.9333
Test Precision (SVM): 0.9333
Test Recall (SVM): 0.9333
Test F1 Score (SVM): 0.9333
Test ROC AUC (SVM): 0.9916
Test Log Loss (SVM): 0.1359
```

```
Test Classification Report (SVM):
              precision    recall  f1-score   support

         0.0       0.93      0.93      0.93        45
         1.0       0.93      0.93      0.93        45

    accuracy                           0.93        90
   macro avg       0.93      0.93      0.93        90
weighted avg       0.93      0.93      0.93        90
```

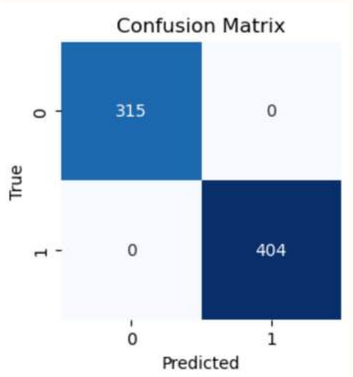


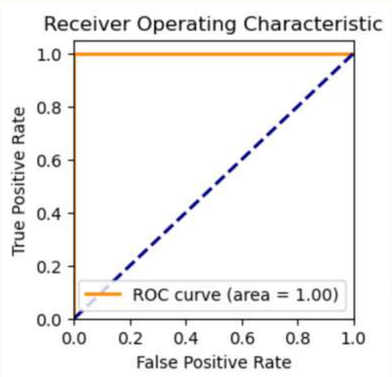


RANDOM FOREST

RANDOM FOREST METRICS FOR BASELINE MODEL

```
Train Entropy: 0.9889189412129864
Train MSE: 0.0
Train Gini Impurity: 0.49233888049582064
Train Log Loss (Cross-Entropy): 0.08043613252447657
Training Accuracy (RF): 1.0000
Training Balanced Accuracy (Rf): 1.0000
Training Precision (RF): 1.0000
Training Recall (RF): 1.0000
Training F1 Score (RF): 1.0000
Training ROC AUC (RF): 1.0000
Training Log Loss (RF): 0.0804
```



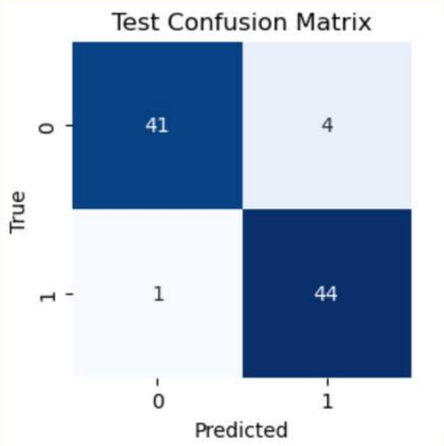


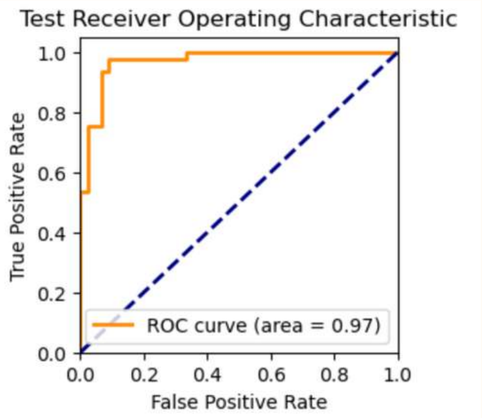
BEST PARAMETERS, VALIDATION ACCURACY OF RF HYPERPARAMETER TUNED MODEL

```
Best Parameters:  {'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimator
s': 100}
Best Accuracy: 0.9430
Validation Accuracy (RF): 0.9333
```

METRICS FOR RF HYPERPARAMETER TUNED WITH TEST DATA

```
Test Entropy: 1.0
Test MSE: 0.05555555555555555
Test Gini Impurity: 0.5
Test Log Loss (Cross-Entropy): 0.1303664399834323
Test Accuracy (RF): 0.9444
Test Balanced Accuracy (RF): 0.9444
Test Precision (RF): 0.9167
Test Recall (RF): 0.9778
Test F1 Score (RF): 0.9462
Test ROC AUC (RF): 0.9719
Test Log Loss (RF): 0.2601

 Test Classification Report (RF):
              precision    recall  f1-score   support

         0.0       0.98      0.91      0.94        45
         1.0       0.92      0.98      0.95        45

    accuracy                           0.94        90
```

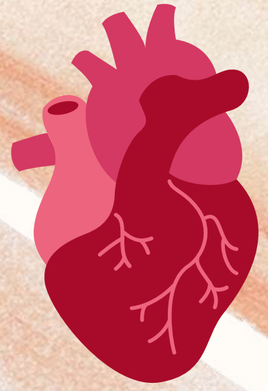




4.5 MODEL VALIDATION AND EVALUATION METHODS

Model	Train Accuracy	Validation Accuracy	Test Accuracy	F1 score	ROC Curve
XGBoost	1.00	0.90	0.8667	0.85	0.94
SVM	1.00	0.9778	0.9333	0.93	0.99
Decision Trees	1.00	0.778	0.83	0.85	0.83
Random Forest	1.00	0.94	0.94	0.94	0.97

- Training Results - Overfitting (as seen in the table)
- Validation Results - Better results with hyperparameter tuning & cross validation
- Final Evaluation on Test Data - Retrained models with Best Parameters



CONCLUSION

Support Vector Machine (SVM) and Random Forest Performance

- SVM demonstrated excellence in accuracy and classification.
- Random Forest showcased robust predictive capabilities with resilience to overfitting.

Enhancements with XGBoost and Decision Trees

- The integration of XGBoost, for efficient gradient boosting.
- Decision Trees were chosen for their simplicity and interoperability.

Holistic Approach

- The combined strengths of SVM, Random Forest, XGBoost, and Decision Trees enhanced our modelling approach for accurate diagnosis.

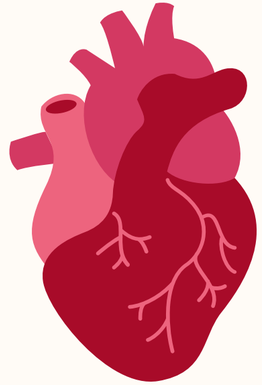
FUTURE SCOPE

- Collection of more diverse data.
- Increase size of dataset (> 899).
- Explore more dimensionality reduction techniques.
- Conduct more intensive hyperparameter tuning.
- Consult medical practitioners to validate diagnosis made by models.

REFERENCES

1. **Mythili, T., Mukherji, D., Padalia, N., & Naidu, A. (2013). *A heart disease prediction model using SVM-decision trees-logistic regression (SDL)*. International Journal of Computer Applications, 68(16). https://www.academia.edu/56949651/A_Heart_Disease_Prediction_Model_using_SVM_Decision_Trees_Logistic_Regression_SDL_**
2. **Ahmad, A. A., & Polat, H. (2023). *Prediction of Heart Disease Based on Machine Learning Using Jellyfish Optimization Algorithm.* Diagnostics, 13(14), 2392. https://www.mdpi.com/2075-4418/13/14/2392**
3. **Patra, R., & Khuntia, B. (2019, February). *Predictive analysis of rapid spread of heart disease with data mining*. In 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-4). IEEE. https://ieeexplore.ieee.org/abstract/document/8869194/**
4. **Khurana, P., Sharma, S., & Goyal, A. (2021, August). Heart disease diagnosis: Performance evaluation of supervised machine learning and feature selection techniques. In 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN) (pp. 510- 515). IEEE. https://ieeexplore.ieee.org/abstract/document/9565963**
5. **Bhatt, C. M., Patel, P., Ghetia, T., & Mazzeo, P. L. (2023). *Effective heart disease prediction using machine learning techniques.* Algorithms, 16(2), 88. https://www.mdpi.com/1999-4893/16/2/88**
6. **Shah, D., Patel, S., & Bharti, S. K. (2020). *Heart disease prediction using machine learning techniques*. SN Computer Science, 1, 1-6. https://link.springer.com/article/10.1007/s42979-020-00365-y**
7. **Senthilkumar Mohan; Chandrasegar Thirumalai; Gautam Srivastava Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques https://ieeexplore.ieee.org/abstract/document/8740989**

LINK TO GITHUB



THANK YOU!

Q&A