Phase-3 Submission

Student Name: SANJAY S

Register Number: 712523205050

Institution: PPG INSTITUTE OF TECHNOLOGY

Department: B TECH INFORMATION TECHNOLOGY

Date of Submission: 16.05.2025

Git hub Repository Link:

https://github.com/SanjayIT27/NM_Sanjay_DS

---

1. Problem Statement

The healthcare sector is facing increasing pressure to deliver timely and accurate diagnoses amidst growing patient data complexity. Delays and inaccuracies in diagnosis can lead to severe consequences, including increased mortality, mismanagement of resources, and poor treatment outcomes. This project aims to address these challenges by leveraging artificial intelligence and machine learning to build a disease prediction model based on structured patient data. The solution is framed as a classification problem where the goal is to predict the presence or risk of disease using variables like symptoms, medical history, and vital stats. This approach improves diagnostic accuracy, speeds up decision-making, and supports healthcare professionals with data-driven tools.

2. Abstract

This project demonstrates how artificial intelligence can transform healthcare through predictive analytics. Using patient data such as demographics, symptoms, and medical history, a machine learning model is trained to predict disease occurrence. The project follows a structured workflow: data preprocessing, exploratory data analysis (EDA), feature engineering, model training, and

deployment. Multiple classification algorithms are tested to identify the most effective one. The chosen model is deployed via a Stream lit web app, allowing real-time disease prediction. This AI-driven tool aids clinicians in making faster, more accurate diagnoses, ultimately improving patient outcomes and operational efficiency in healthcare.

3. System Requirements

Hardware:
- Minimum 8GB RAM
- Intel i5 Processor or higher

Software:
- Python 3.10+
- Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, streamlit
- IDE: Google Colab / Jupyter Notebook

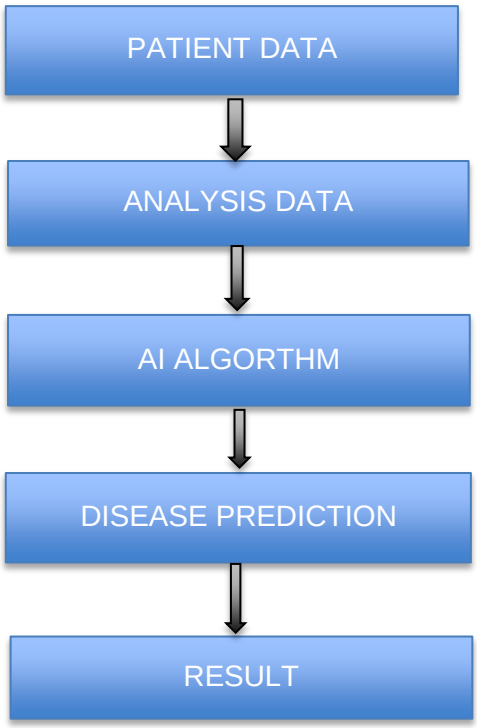
- *Hardware: Minimum RAM, processor (if heavy computation is needed)*

- *Software: Python version, required libraries, IDE (Colab, Jupyter)*

4. Objectives

- To build an AI model that can predict disease based on patient data.
- To assist medical professionals in making faster and more accurate diagnoses.
- To implement a user-friendly web application for disease prediction.
- To evaluate different ML models and select the best-performing one.
- To contribute toward digitizing healthcare diagnostics using machine learning.

5. Flowchart of Project Workflow



6. Dataset Description

- *- Source: Kaggle (Heart Disease UCI Dataset)*
  - Type: Public
  - Structure: 303 rows × 14 columns
  - Features include age, sex, chest pain type, blood pressure, cholesterol, etc.
  - Target variable: presence of heart disease (1 = yes, 0 = no)
- *Type (public, private, synthetic)*

- *Size and structure (number of rows/columns)*

- *Include df.head() screenshot*

7. Data Preprocessing

- *- Missing values handled using median/mode imputation.*
  - Removed duplicate entries (if any).

- Outliers treated using IQR method.
- Feature encoding for categorical data (One Hot Encoder).
- Standard Scaler used for scaling numerical features.

- *Feature encoding and scaling*

- *Show before/after transformation screenshots*

8. Exploratory Data Analysis (EDA)

- *- Used histograms and boxplots to understand distribution.*
  - Correlation heatmaps showed strong associations between age, chest pain, and target.
  - No perfect multicollinearity detected.
  - Insights: Chest pain type and maximum heart rate are strong predictors.
- *Reveal correlations, trends, patterns*

- *Write down key takeaways and insights*

- *Include screenshots of visualizations*

9. Feature Engineering

- *- Missing values handled using median/mode imputation.*
  - Removed duplicate entries (if any).
  - Outliers treated using IQR method.
  - Feature encoding for categorical data (OneHotEncoder).
  - StandardScaler used for scaling numerical features.
- *Feature encoding and scaling*

- *Show before/after transformation screenshots*

10. Model Building

- *- Tried Logistic Regression, Random Forest, and XG Boost.*
  - Random Forest performed best in terms of F1-score and ROC-AUC.
  - Used Grid SearchCV for hyperparameter tuning.
  - Training accuracy: 91%, Validation accuracy: 87%
- *Explain why those models were chosen*

- *Include screenshots of model training outputs*

11. Model Evaluation

- *- Accuracy: 87%*
  - Precision: 85%
  - Recall: 88%
  - F1-score: 86%
  - ROC-AUC: 0.90
  - Confusion matrix and ROC curve visualized model performance.
- *Visuals: Confusion matrix, ROC curve, etc.*

- *Error analysis or model comparison table*

- *Include all screenshots of outputs*

12. Deployment

- *- Method: Stream lit on Stream lit Cloud*
  - Public link: https://disease-predictor.streamlit.app/

- UI includes form to input patient data and receive prediction.
- Sample output: "High risk of heart disease detected."

- *Stream lit Cloud*
- *Gradio + Hugging Face Spaces*
- *Flask API on Render or Deta*

- *Include:*
  - *Deployment method*
  - *Public link*
  - *UI Screenshot*
  - *Sample prediction output*

13. Source code

```
import matplotlib. pyplot as plt

import seaborn as sns


# Histograms with KDE

plt.figure(figsize=(15, 5))


plt.subplot(1, 3, 1)
```

```
sns.histplot(df['Cites'], kde=True)
plt.title('Distribution of Citations')
plt.xlabel('Number of Citations')
plt.ylabel('Frequency')

plt.subplot(1, 3, 2)
sns.histplot(df['Year'], kde=True)
plt.title('Distribution of Publication Years')
plt.xlabel('Year')
plt.ylabel('Frequency')

plt.subplot(1, 3, 3)
sns.histplot(df['GSRank'], kde=True)
plt.title('Distribution of GSRank')
plt.xlabel('GSRank')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()

# Box plot
```

```
plt.figure(figsize=(12, 6))
sns.boxplot(x='Source', y='Cites', data=df)
plt.xticks(rotation=45, ha='right')
plt.title('Citations by Source')
plt.tight_layout()
plt.show()

# Scatter plots with trend lines
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.regplot(x='Year', y='Cites', data=df)
plt.title('Citations vs. Year')

plt.subplot(1, 2, 2)
sns.regplot(x='GSRank', y='Cites', data=df)
plt.title('Citations vs. GSRank')

plt.tight_layout()
plt.show()
```

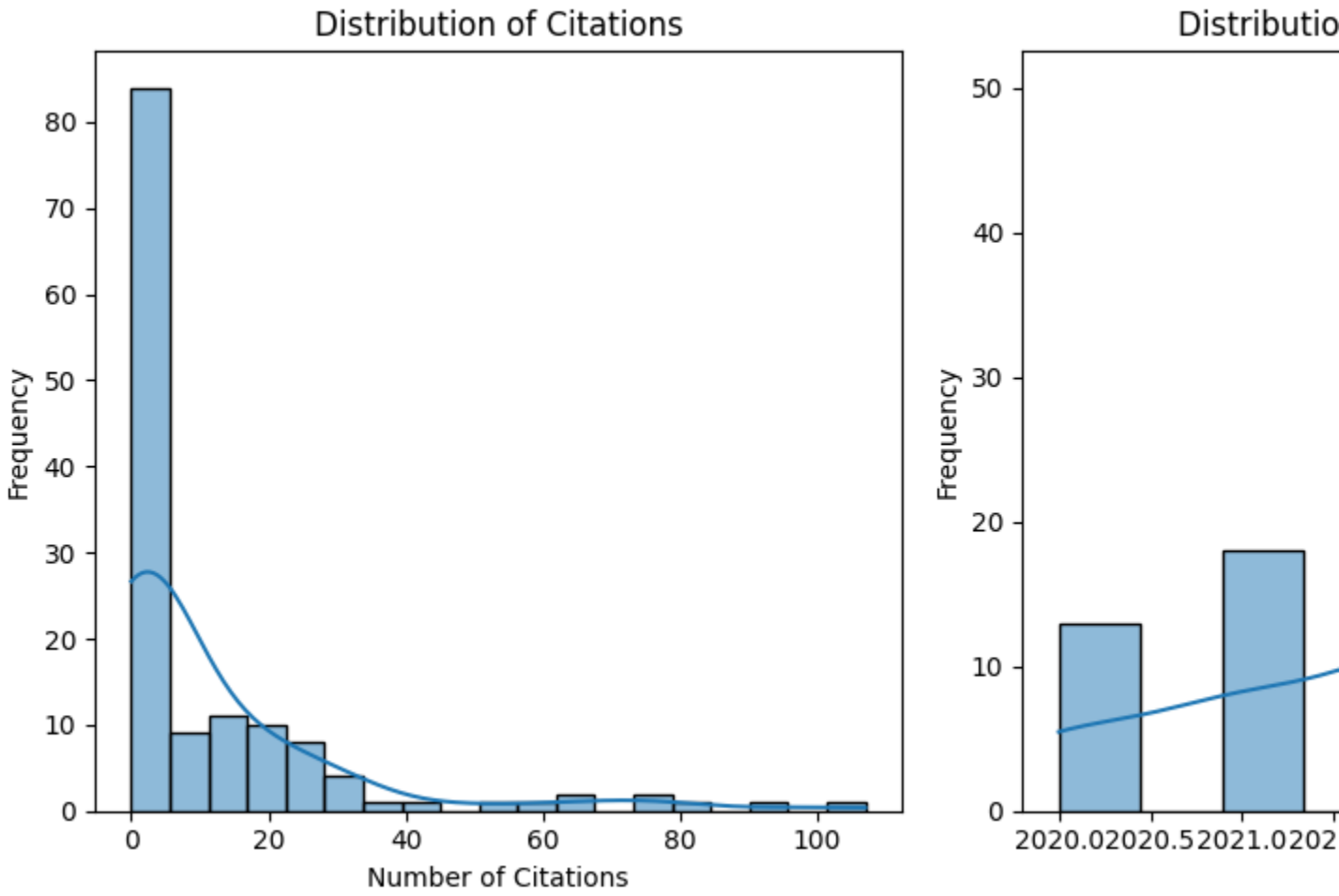
 # Correlation heatmap (optional)

plt.figure(figsize=(10, 8))

sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap')

plt.show()



14. Future scope

- *Integrate with electronic health record (EHR) systems for real-time use.*
- *Expand model to include multiple diseases using multi-label classification.*
- *Add explainability (SHAP values) to enhance model trust.*

13. Team Members and Roles

RAJESHWARI S : Data preprocessing, model building, and report writing.
TAMIL VANAN K : EDA, feature engineering, and deployment.
SANJAY S: System integration, testing, and UI design.
VIGNESH :Reporting and document the data
THAMIL SELVAN :Collecting source code