SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

TAMIL NADU, COIMBATORE

Department of Artificial Intelligence and Machine Learning

INTERNSHIP REPORT
AT Unified Mentor

Report Submitted By

Name : **Hareesh kumar K**

Class : **III Year B.Tech., AIML**

Academic Year : **2025 – 2026**

Duration : **15.05.2025 – 15.08.2025**

ACKNOWLEDGEMENT

I am deeply honored to express my heartfelt gratitude to **Unified Mentor** for granting me the opportunity to undertake this enriching internship. The organization provided an exceptional platform that not only enhanced my technical and professional skills but also fostered an environment of learning, collaboration, and innovation. This internship has been a pivotal milestone in my career journey, and I am immensely grateful for the trust and support extended to me throughout this period.

I extend my sincere appreciation to my supervisor whose exemplary guidance, insightful feedback, and unwavering encouragement were instrumental in shaping the outcome of my projects. Their expertise, patience, and commitment to nurturing my skills helped me navigate challenges and achieve the objectives of the internship with confidence and clarity.

I am equally thankful to my colleagues and team members at Unified Mentor. Their collaborative spirit, willingness to share knowledge, and constant support created a dynamic and inspiring work environment. The synergy within the team played a crucial role in the successful completion of all four of my internship projects, and I am grateful for their camaraderie and contributions.

I would also like to acknowledge the contributions of any external mentors, advisors, or institutional collaborators who provided valuable insights and expertise during the course of this internship. Their input enriched my understanding and added depth to my work.

Lastly, I owe a special debt of gratitude to my friends and family, whose unconditional support, encouragement, and belief in my abilities kept me motivated throughout this journey. Their presence provided me with the emotional strength to overcome challenges and stay focused on my goals.

This internship experience has been transformative, and I carry forward the lessons, skills, and connections gained at Unified Mentor with immense gratitude and pride.

INTERNSHIP DETAILS

Name of the Industry	**:**	Unified Mentor
Address	**:**	Cyber City, WeWork DLF Forum, Haryana 122002
Website	**:**	https://www.unifiedmentor.com/
Internship Duration	**:**	3 Months
Contact No	**:**	+91 08645322947
E-mail id	**:**	hello@unifiedmentor.com

INTRODUCTION

During my internship at Unified Mentor, I embarked on a pivotal project titled **"Detect Heart Disease Using Patient Data"**, a machine learning-driven initiative aimed at predicting the likelihood of heart disease in patients using a combination of clinical and lifestyle parameters. This project is a response to the increasing need for data-driven healthcare solutions that empower healthcare professionals and individuals with accessible, reliable, and efficientGarrett, I aim to create a robust and accurate predictive model that can be deployed through an interactive web application, ensuring both precision in predictions and ease of use for end-users. The primary objective of this project was to develop a comprehensive end-to-end solution, from data collection and preprocessing to model training, evaluation, and deployment, to facilitate preliminary heart disease risk assessments.

This project was undertaken as part of my internship to address a critical healthcare challenge: early detection of heart disease, which remains a leading cause of mortality worldwide. By leveraging machine learning, the project sought to provide a tool that could assist in identifying at-risk individuals based on readily available clinical data, such as age, sex, cholesterol levels, and other physiological indicators. The solution was designed to be both technically robust and user-friendly, ensuring that it could be adopted by healthcare providers or individuals with minimal technical expertise. The project involved a meticulous process of data preprocessing, exploratory data analysis (EDA), feature engineering, model development, and deployment via a Streamlit-based web application. This report provides a detailed account of the project's methodology, implementation challenges, results, and the key learnings gained throughout the process, offering a comprehensive overview of the end-to-end workflow.

The dataset used in this project consisted of 1190 patient records, each containing 10 key features, including sex, chest pain type, fasting blood sugar, resting electrocardiogram (ECG) results, exercise-induced angina, oldpeak (ST depression), ST slope, cholesterol-to-blood-pressure ratio, age multiplied by maximum heart rate, and the target variable indicating the presence or absence of heart disease. The project culminated in the development of a Random Forest Classifier, selected for its interpretability and robustness, achieving an impressive accuracy of 92.86%. The deployment of the model through a Streamlit application further ensured that the predictive tool was accessible and practical for real-world use.

PROJECT OVERVIEW

The Heart Disease Prediction project, undertaken during my internship at Unified Mentor, aimed to develop a sophisticated machine learning-based solution to classify patients as either having heart disease or not, based on a comprehensive dataset of clinical and lifestyle parameters. This initiative was driven by the critical need to address heart disease, a leading global cause of mortality, by providing a reliable, data-driven tool for preliminary risk assessment. The project targeted healthcare professionals seeking efficient diagnostic aids and individuals interested in understanding their cardiovascular risk profile. By integrating advanced machine learning techniques with an accessible user interface, the project sought to bridge the gap between complex data analytics and practical healthcare applications.

The core of the project revolved around a meticulously curated dataset comprising 1190 patient records, each with 10 features: sex, chest pain type, fasting blood sugar, resting electrocardiogram (ECG) results, exercise-induced angina, oldpeak (ST depression), ST slope, cholesterol-to-blood-pressure ratio, age multiplied by maximum heart rate, and a binary target variable indicating the presence (1) or absence (0) of heart disease. The dataset was carefully preprocessed to ensure high data quality, addressing issues such as missing values, inconsistencies, and feature scaling to optimize model performance.

The project employed a Random Forest Classifier as the primary predictive model, selected for its robustness, ability to handle mixed feature types (categorical and numerical), resistance to overfitting, and interpretability through feature importance scores. This choice was informed by a comparative evaluation of multiple algorithms, including Support Vector Classifier (SVC), Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree Classifier, with Random Forest outperforming others in accuracy and reliability. To enhance the model's predictive power, feature engineering was conducted to derive two novel features: the cholesterol-to-blood-pressure ratio (cholesterol_bp), which captures cardiovascular risk, and age multiplied by maximum heart rate (age_max_hr), a composite metric reflecting physiological stress. These engineered features were designed to incorporate domain-specific insights, improving the model's ability to detect subtle patterns indicative of heart disease.

The final model was rigorously trained and evaluated, achieving an impressive accuracy of 92.86%, alongside strong F1, recall, and ROC AUC scores, ensuring both high sensitivity and

specificity in predictions. To make the model accessible to end-users, it was seamlessly integrated into a Streamlit web application, developed in the app.py file. This application provides an intuitive, interactive interface that allows users to input patient data and receive real-time predictions, complete with probability scores for both heart disease and no heart disease outcomes. The application was designed with usability in mind, featuring clear input fields for clinical parameters and immediate, interpretable outputs, making it suitable for healthcare professionals, researchers, and individuals with minimal technical expertise.

The project's success was measured by several key metrics: the high predictive accuracy of the model, the seamless functionality of the Streamlit application, and the clarity of the user experience. The application ensures that users can input data such as sex, chest pain type, and other clinical metrics, which are then processed using the saved Random Forest model (rf_model.pkl) and a standardized scaler (rf_scl.pkl) to ensure consistency with the training phase. The output not only provides a binary prediction (heart disease or no heart disease) but also displays the probability of each class, enhancing transparency and trust in the results.

The project was structured to ensure scalability and reproducibility, with a well-organized file structure that includes:

- cleaned_data.csv: The preprocessed dataset used for training.
- int_ht.csv: An intermediate dataset generated during feature engineering.
- EDA_FE.ipynb: A Jupyter notebook documenting exploratory data analysis and feature engineering.
- main.ipynb: A notebook detailing model training, evaluation, and hyperparameter tuning.
- app.py: The Streamlit application for real-time predictions.
- rf_model.pkl: The saved Random Forest model.
- rf_scl.pkl: The saved StandardScaler for consistent preprocessing.

This structured approach ensures that the project can be easily understood, maintained, or extended by other developers or researchers. The Heart Disease Prediction project not only demonstrates the power of machine learning in healthcare but also serves as a practical tool for preliminary risk assessment, with potential applications in clinical settings, health awareness campaigns, or further research into cardiovascular health. The combination of a high-performing model, thoughtful feature engineering, and a user-friendly deployment platform underscores the project's success in meeting its objectives

METHODOLOGY

1. Data Collection and Preprocessing

The foundation of the Heart Disease Prediction project was a carefully curated dataset stored in cleaned_data.csv, comprising 1190 patient records, each with 10 features, including a binary target variable indicating the presence (1) or absence (0) of heart disease. The features included sex, chest pain type, fasting blood sugar, resting electrocardiogram (ECG) results, exercise-induced angina, oldpeak (ST depression), ST slope, cholesterol-to-blood-pressure ratio (cholesterol_bp), and age multiplied by maximum heart rate (age_max_hr). To ensure the dataset was suitable for machine learning, a comprehensive preprocessing pipeline was implemented, addressing data quality, encoding, feature engineering, and scaling. The preprocessing steps were as follows:

- Data Cleaning: The dataset was thoroughly inspected to identify and address data quality issues. Null values, which could introduce bias or errors in model predictions, were removed. Inconsistent entries, such as outliers beyond physiological norms (e.g., unrealistic cholesterol levels or heart rates), were corrected or excluded to maintain data integrity. Duplicate records were also eliminated to prevent overfitting during model training. This cleaning process ensured a robust and reliable dataset for subsequent analysis and modeling.
- Feature Encoding: Several features in the dataset, such as sex (Male/Female) and exercise-induced angina (Yes/No), were categorical in nature. To make these variables compatible with machine learning algorithms, they were encoded into numerical formats. Specifically, sex was encoded as Male = 1 and Female = 0, while exercise-induced angina was encoded as Yes = 1 and No = 0. Other categorical features, such as chest pain type (values 1–4) and ST slope (values 0–2), were already in numerical form but were validated to ensure consistency and correctness.
- Feature Engineering: To enhance the model's predictive power, two novel features were derived based on domain knowledge of cardiovascular health:
    - Cholesterol/BP Ratio (cholesterol_bp): This feature was calculated as the ratio of a patient's cholesterol level to their blood pressure. Elevated cholesterol relative to blood pressure is a known indicator of cardiovascular risk, as it reflects potential arterial strain and plaque buildup. This engineered feature

provided a composite metric to capture this relationship, improving the model's ability to detect heart disease risk.

    - Age × Max Heart Rate (age_max_hr): This composite feature was created by multiplying a patient's age by their maximum heart rate, reflecting physiological stress and cardiovascular capacity. Older individuals with higher maximum heart rates may exhibit different risk profiles compared to younger individuals, and this feature helped capture such interactions. These engineered features were carefully validated to ensure they added meaningful predictive value without introducing redundancy.

- Scaling: To ensure that features with different scales (e.g., age_max_hr with large values versus oldpeak with smaller values) did not disproportionately influence the model, all numerical features were standardized using the StandardScaler from scikit-learn. This process transformed the features to have a mean of 0 and a standard deviation of 1, normalizing their distributions and improving model performance, particularly for algorithms sensitive to feature scales, such as Support Vector Machines and K-Nearest Neighbors. The trained scaler was saved as rf_scl.pkl to ensure consistent preprocessing during model deployment, allowing the Streamlit application to apply the same transformations to user inputs.
- Intermediate Dataset: During the feature engineering process, an intermediate dataset named int_ht.csv was generated to store partially processed data. This file served as a checkpoint, allowing iterative experimentation with feature engineering techniques before finalizing the cleaned and engineered dataset in cleaned_data.csv. The intermediate dataset facilitated collaboration and ensured that preprocessing steps could be revisited without altering the original data.

These preprocessing steps were critical to preparing a high-quality dataset that maximized the model's ability to learn meaningful patterns while minimizing noise and bias. The cleaned and engineered dataset formed the foundation for subsequent exploratory data analysis and model development.

EXPLORATORY DATA ANALYSIS

Exploratory Data Analysis (EDA) was conducted using the Jupyter notebook EDA_FE.ipynb to gain a deep understanding of the dataset's characteristics, identify patterns, and inform feature selection and engineering decisions. The EDA process was a critical step in ensuring that the dataset was well-suited for modeling and that the selected features were both relevant and predictive. The key activities performed during EDA included:

- Feature Distribution Analysis: Each feature was visualized using histograms and box plots to assess its distribution, identify skewness, and detect outliers. For example, continuous features like cholesterol_bp and age_max_hr were examined to ensure their values fell within expected physiological ranges, while categorical features like chest pain type and ST slope were analyzed to confirm their category distributions. This step helped identify potential data quality issues, such as extreme outliers, which were addressed during preprocessing.
- Correlation Analysis: A correlation matrix was generated using seaborn's heatmap functionality to evaluate relationships between features. This analysis helped identify potential multicollinearity, where highly correlated features could reduce model interpretability or performance. For instance, correlations between cholesterol_bp and other cardiovascular indicators were scrutinized to ensure the engineered feature provided unique predictive value. Features with excessive correlation were considered for removal to streamline the dataset.
- Class Balance Verification: The distribution of the target variable (heart disease vs. no heart disease) was analyzed to ensure a balanced dataset. Imbalanced classes could lead to biased models that favor the majority class, so this step confirmed that the dataset had a roughly equitable split between positive (heart disease) and negative (no heart disease) cases. Techniques such as stratified sampling were used during train-test splitting to maintain this balance in both training and testing sets.
- Feature Importance Exploration: Preliminary models, such as a basic Random Forest Classifier, were trained to estimate feature importance scores. This analysis helped identify which features, including the engineered cholesterol_bp and age_max_hr, contributed most significantly to predicting heart disease. Insights from this step guided the retention of high-impact features and the elimination of redundant or low-value ones, optimizing the dataset for model training.

The EDA process not only validated the quality of the preprocessed dataset but also provided actionable insights that shaped the feature engineering and model development phases. By understanding the data's structure and relationships, the project ensured that the subsequent modeling efforts were grounded in a robust and informed foundation.

MODEL DEVELOPMENT

The model development phase, documented in the main.ipynb Jupyter notebook, involved a systematic approach to selecting, training, and evaluating machine learning models to identify the most effective algorithm for heart disease prediction. The goal was to develop a model that balanced accuracy, interpretability, and robustness, suitable for deployment in a healthcare context. The process included model selection, hyperparameter tuning, performance evaluation, and model serialization for deployment. The key steps were as follows:

- Model Selection: Five machine learning algorithms were evaluated to determine the best fit for the dataset:
    - Support Vector Classifier (SVC): A powerful algorithm for high-dimensional data, effective for binary classification but computationally intensive for large datasets.
    - Logistic Regression: A simple and interpretable linear model, well-suited for binary classification tasks but potentially limited in capturing complex relationships.
    - K-Nearest Neighbors (KNN): A non-parametric algorithm that classifies based on feature similarity, effective for small datasets but sensitive to feature scaling.
    - Decision Tree Classifier: A tree-based model that is highly interpretable but prone to overfitting without proper pruning or tuning.
    - Random Forest Classifier: An ensemble method that combines multiple decision trees to improve robustness and reduce overfitting, ideal for handling mixed feature types and providing feature importance scores.

Each model was trained on the preprocessed dataset, with the training set comprising 80% of the data (952 records) and the test set comprising 20% (238 records), split using train_test_split with a random state of 42 for reproducibility. The models were evaluated using multiple metrics: accuracy, F1 score, recall, and ROC AUC score, to ensure a comprehensive assessment of performance.

- Model Performance and Selection: The Random Forest Classifier outperformed the other models, achieving superior accuracy and robustness due to its ensemble nature and ability to handle both categorical and numerical features effectively. Its resistance to overfitting and ability to provide feature importance scores made it particularly

suitable for a healthcare application, where interpretability is crucial. The initial performance metrics for the Random Forest Classifier on the test set were promising, prompting further optimization through hyperparameter tuning.

- Hyperparameter Tuning: To maximize the Random Forest Classifier's performance, hyperparameter tuning was conducted using GridSearchCV with a 5-fold cross-validation strategy. The parameter grid included:
    - max_depth: [5, 8, 15, None, 10] – to control tree depth and prevent overfitting.
    - max_features: [5, 7, 'auto', 8] – to determine the number of features considered at each split.
    - min_samples_split: [2, 8, 12, 15, 20] – to set the minimum number of samples required to split a node.
    - min_samples_leaf: [1, 2, 4, 6, 8, 10] – to set the minimum number of samples at a leaf node.
    - n_estimators: [200, 300, 500, 1000] – to specify the number of trees in the forest.
    - bootstrap: [True, False] – to enable or disable bootstrapping for tree sampling.

The tuning process identified the optimal parameters:

- n_estimators: 300
- min_samples_split: 2
- max_features: 5
- max_depth: 15
- bootstrap: True
- min_samples_leaf: 1

These parameters balanced model complexity and generalization, resulting in a highly accurate and stable model.

- Final Model Performance: The tuned Random Forest Classifier was retrained on the entire training set and evaluated on the test set, achieving the following metrics:
    - Accuracy: 0.9286 (92.86%), indicating that 92.86% of predictions were correct.
    - F1 Score: 0.9281, reflecting a balanced trade-off between precision and recall.
    - Recall Score: 0.9771, demonstrating high sensitivity in identifying heart disease cases, critical for a healthcare application where missing positive cases is costly.
    - ROC AUC Score: 0.9231, indicating strong discrimination between heart disease and no heart disease classes.

These metrics confirmed the model's reliability and suitability for deployment. The trained model was serialized using joblib and saved as rf_model.pkl for use in the Streamlit application.

APPLICATION DEVELOPMENT

To translate the predictive model into a practical tool, a web application was developed using Streamlit, a Python framework for building interactive data applications. The application, implemented in the app.py file, was designed to provide a user-friendly interface for healthcare professionals and individuals to input patient data and receive real-time predictions. The application's development focused on usability, accuracy, and seamless integration with the trained model. The key components of the application were:

- User Interface: The Streamlit application featured an intuitive interface with input fields for all 9 predictive features:
    - Sex: A select box allowing users to choose between Male and Female.
    - Chest Pain Type: A select box with options 1 to 4, representing different types of chest pain (e.g., typical angina, atypical angina, non-anginal pain, asymptomatic).
    - Fasting Blood Sugar: A number input field where users enter a value, automatically encoded as 1 if >120 mg/dL or 0 if ≤120 mg/dL.
    - Resting ECG Results: A number input field accepting values 0 (normal), 1 (ST-T wave abnormality), or 2 (probable/definite left ventricular hypertrophy).
    - Exercise-Induced Angina: A select box for Yes or No, indicating whether angina is induced by exercise.
    - Oldpeak: A number input for ST depression values, typically ranging from 0.0 (normal) to higher values indicating risk.
    - ST Slope: A number input for values 0 (upsloping/normal), 1 (flat/risk), or 2 (downsloping/high risk).
    - Cholesterol/BP Ratio: A number input for the derived cardiovascular risk metric, with guidelines provided (e.g., <1.3 for low risk, >2.0 for high risk).
    - Age × Max Heart Rate: A number input for the composite physiological metric, with ranges indicating risk levels (e.g., <3000 for very low, 3000–9000 for average to high).

Each input field was accompanied by descriptive labels to guide users, ensuring accessibility for non-technical audiences.

- Input Processing: Upon submission via a "Make Prediction" button, the application

processed the inputs as follows:

  - Encoding: Categorical inputs (e.g., sex, exercise-induced angina) were converted to numerical values (e.g., Male = 1, Female = 0, Yes = 1, No = 0).
  - Thresholding: Fasting blood sugar was encoded as 1 if the input value exceeded 120 mg/dL, otherwise 0.
  - Scaling: All numerical inputs were transformed using the saved StandardScaler (rf_scl.pkl) to match the preprocessing applied during model training, ensuring consistency and accuracy in predictions.
  - Prediction: The processed input was passed to the saved Random Forest model (rf_model.pkl), which generated a binary prediction (0 for no heart disease, 1 for heart disease) and probability scores for both classes.

- Output Display: The application presented the prediction results in a clear and visually appealing format:
  - If the prediction was 1, a red "error" message indicated that the patient was likely to have heart disease, with an emoji.
  - If the prediction was 0, a green "success" message indicated that the patient was likely normal (no heart disease), with a checkmark emoji
  - The probability scores for both classes (e.g., [No Heart Disease: 0.62, Heart Disease: 0.38]) were displayed to provide transparency and context, allowing users to gauge the model's confidence in its prediction.

The Streamlit application was tested locally to ensure functionality, with inputs validated to prevent errors (e.g., ensuring numerical inputs fell within acceptable ranges). The use of saved model and scaler files ensured that the application was lightweight and portable, ready for potential cloud deployment or integration into other platforms.

RESULTS

The Heart Disease Prediction project delivered a robust and practical solution, achieving the following outcomes:

- Model Performance: The Random Forest Classifier achieved an accuracy of 92.86% on the test set, with an F1 score of 0.9281, a recall score of 0.9771, and a ROC AUC score of 0.9231. These metrics indicate a highly reliable model with strong sensitivity, particularly important for identifying heart disease cases to minimize false negatives in a healthcare context.
- Deployment Success: The Streamlit application was fully functional, enabling real-time predictions with a seamless user experience. The interface was designed to be intuitive, with clear input fields and immediate, interpretable outputs, making it accessible to healthcare professionals, researchers, and individuals without technical expertise.
- Scalability and Reusability: The project's components, including the saved model (rf_model.pkl) and scaler (rf_scl.pkl), were serialized using joblib, ensuring that the solution could be easily integrated into other systems or extended for future enhancements. The modular design supports scalability, such as deploying the application on a cloud platform or incorporating additional features.
- Organized File Structure: The project was structured for clarity and reproducibility, with the following components:
    - app.py: The Streamlit application for real-time predictions.
    - main.ipynb: Jupyter notebook for model training, evaluation, and hyperparameter tuning.
    - EDA_FE.ipynb: Jupyter notebook for exploratory data analysis and feature engineering.
    - cleaned_data.csv: The preprocessed dataset used for training.
    - int_ht.csv: An intermediate dataset generated during feature engineering.
    - rf_model.pkl: The serialized Random Forest model.
    - rf_scl.pkl: The serialized StandardScaler for preprocessing.

These results underscore the project's success in delivering a high-performing, user-friendly, and scalable solution for heart disease prediction, with potential applications in clinical diagnostics, health awareness campaigns, and medical research.

CHALLENGES

The development of the Heart Disease Prediction project presented several challenges that required careful consideration and problem-solving:

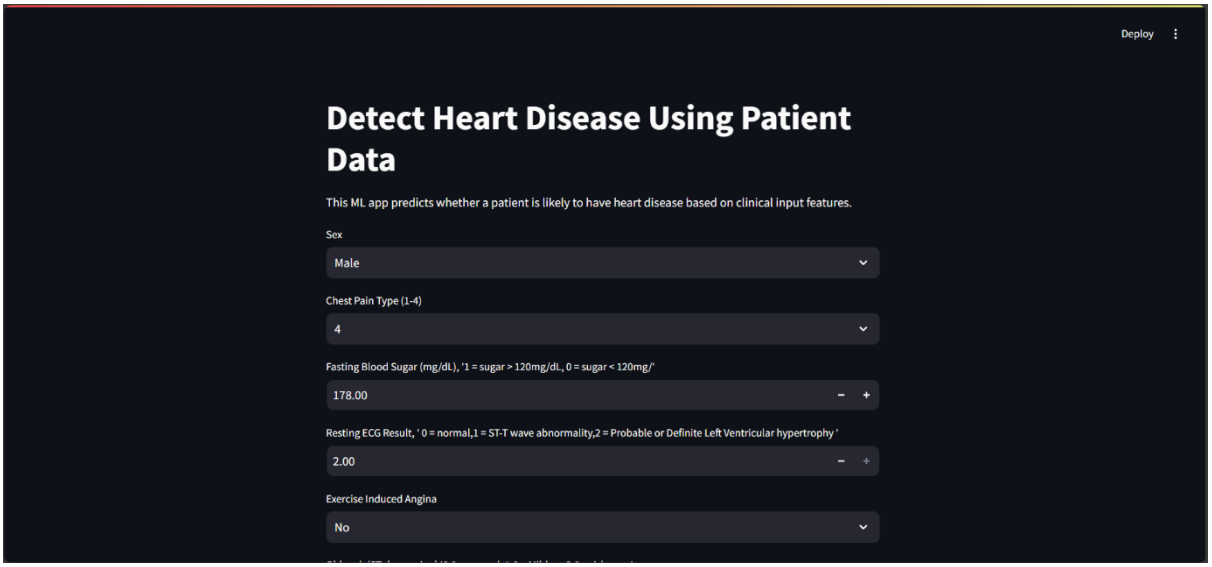
- Data Quality: Ensuring the dataset was free from null values, outliers, and inconsistencies was a significant challenge. For example, physiological outliers (e.g., unrealistic heart rate values) required validation against medical norms, and missing values needed imputation or removal without introducing bias.
- Feature Engineering: Creating medically relevant features like cholesterol_bp and age_max_hr demanded a deep understanding of cardiovascular risk factors. Ensuring these features added predictive value without redundancy required iterative experimentation and validation.
- Hyperparameter Tuning: The computational cost of GridSearchCV for the Random Forest Classifier was substantial, given the large parameter grid and dataset size. Balancing tuning thoroughness with computational efficiency was a key challenge, addressed by carefully selecting parameter ranges and leveraging parallel processing (n_jobs=-2).
- Deployment Consistency: Ensuring that preprocessing steps (e.g., scaling and encoding) were identical between the training phase (in main.ipynb) and the deployment phase (in app.py) was critical to avoid discrepancies in predictions. This required meticulous management of the StandardScaler and validation of input processing logic in the Streamlit application.
- User Interface Design: Designing a Streamlit interface that was both intuitive and robust for non-technical users, such as healthcare professionals, required careful consideration of input formats, error handling, and output clarity. For example, providing descriptive labels and guidelines for inputs like cholesterol_bp ensured users could interpret and enter data correctly.

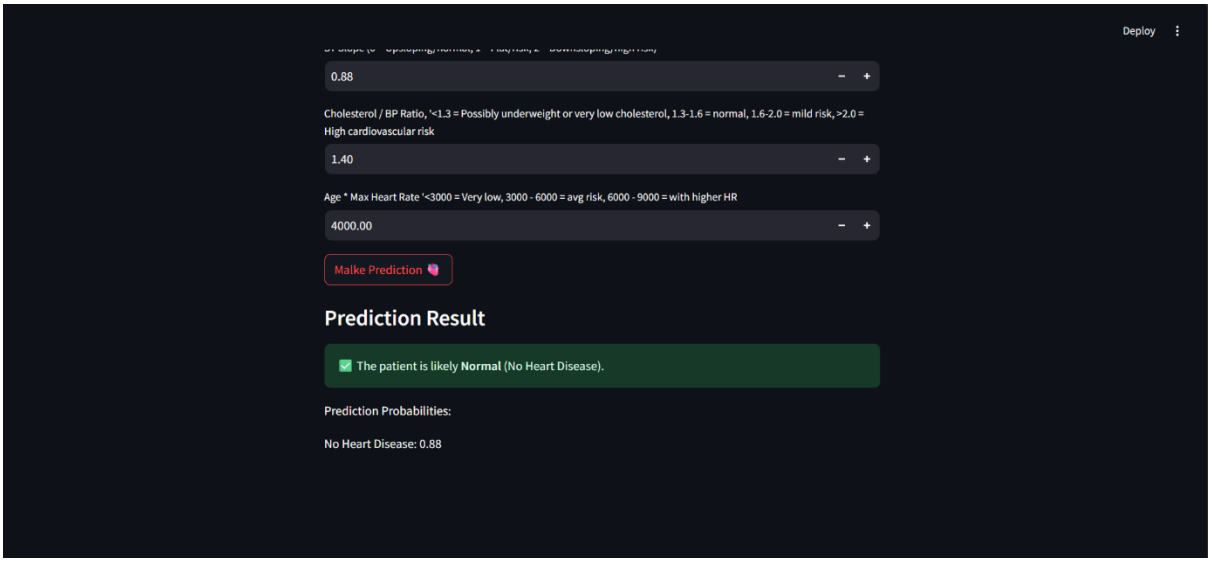
CONCLUSION

The Heart Disease Prediction project, completed during my internship at Unified Mentor, successfully delivered a high-performing, practical, and user-friendly solution for predicting heart disease risk. By leveraging a Random Forest Classifier, the project achieved an impressive accuracy of 92.86%, with strong recall (0.9771) and ROC AUC (0.9231) scores, ensuring reliable and sensitive predictions critical for healthcare applications. The deployment of the model through a Streamlit web application, accessible via app.py, provided an intuitive interface for real-time predictions, making the tool valuable for healthcare professionals, researchers, and individuals seeking preliminary risk assessments.

The project's end-to-end workflow—from data preprocessing and feature engineering to model development and deployment—demonstrated the power of machine learning in addressing real-world healthcare challenges. Key achievements included the creation of medically relevant features (cholesterol_bp and age_max_hr), rigorous model evaluation, and a scalable application design. The organized file structure, including cleaned_data.csv, int_ht.csv, EDA_FE.ipynb, main.ipynb, rf_model.pkl, and rf_scl.pkl, ensures that the project is well-documented and extensible for future enhancements.

The internship experience provided invaluable learnings in data science, web development, and project management, equipping me with the skills to tackle complex problems and deliver impactful solutions. The Heart Disease Prediction project not only highlights the potential of AI in improving healthcare outcomes but also serves as a foundation for future innovations, such as cloud-based deployment, integration of additional clinical features, or adaptation for other medical prediction tasks. This project stands as a testament to the transformative potential of data-driven healthcare and my growth as a data science professional during my time at Unified Mentor.


Deploy
Detect Heart Disease Using Patient Data
This ML app predicts whether a patient is likely to have heart disease based on clinical input features.
Sex
Male
Chest Pain Type (1-4)
4
Fasting Blood Sugar (mg/dL), '1 = sugar > 120mg/dL, 0 = sugar < 120mg/'
178.00
Resting ECG Result, ' 0 = normal,1 = ST-T wave abnormality,2 = Probable or Definite Left Ventricular hypertrophy '
2.00
Exercise Induced Angina
No


Deploy
0.88
Cholesterol / BP Ratio, '<1.3 = Possibly underweight or very low cholesterol, 1.3-1.6 = normal, 1.6-2.0 = mild risk, >2.0 = High cardiovascular risk
1.40
Age * Max Heart Rate '<3000 = Very low, 3000 - 6000 = avg risk, 6000 - 9000 = with higher HR
4000.00
Malke Prediction
Prediction Result
The patient is likely Normal (No Heart Disease).
Prediction Probabilities:
No Heart Disease: 0.88