IMPROVING CLASSIFICATION ACCURACY FOR THE EFFECTIVE DIAGNOSIS OF DISEASES USING HYBRID MODEL

Project submitted in partial fulfilment to the requirement for the award degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Submitted by

SAMUEL V

24-PCS-023

Under the guidance of

Dr. S. BHARATHIDASON, M.Sc., M.Phil., MBA, Ph. D

DEPARTMENT OF COMPUTER SCIENCE

LOYOLA COLLEGE (AUTONOMOUS)

CHENNAI–34

OCTOBER–2025

DECLARATION

I, **SAMUEL V (24-PCS-023)** hereby declare that the project report entitled **"IMPROVING CLASSIFICATION ACCURACY FOR THE EFFECTIVE DIAGNOSIS OF DISEASES USING HYBRID MODEL"** is done under the guidance of **Dr. S. BHARATHIDASON** M.Sc., M.Phil., MBA, Ph. D., at PG Computer lab, Loyola College (Autonomous), Chennai-34 is being submitted in partial fulfilment of the requirements for the award of the degree in **MASTER OF SCIENCE IN COMPUTER SCIENCE.**

DATE:

PLACE: CHENNAI SIGNATURE OF THE STUDENT

BONAFIDE CERTIFICATE

This is to certify that the project work entitled **"IMPROVING CLASSIFICATION ACCURACY FOR THE EFFECTIVE DIAGNOSIS OF DISEASES USING HYBRID MODEL "** is being submitted to Loyola College (Autonomous), Chennai-600034 by SAMUEL V (24-PCS-023) for the partial fulfillment for the award of degree of Master of Science in Computer Science is a Bonafide record of work carried out by her, under my guidance and supervision.

HEAD OF THE DEPARTMENT
Dr. J. Jerald Inico M.Sc., M.Phil., MCP., Ph.D.

PROJECT GUIDE
Dr. S. Bharathidason M.Sc., M.Phil., MBA, Ph.D.

The Viva-Voce Examination held on _____ at Loyola College Autonomous), Chennai – 600034.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

The accurate and timely prediction of diseases is a critical challenge in the healthcare domain. In this work, a performance analysis of multiple machine learning algorithms is carried out for the prediction of five diseases: Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer. The datasets were preprocessed to handle missing values, encode categorical attributes, and normalize numerical features. Four base classifiers Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Extreme Gradient Boosting (XGBoost) were tuned using Grid Search and subsequently integrated into a stacking ensemble with Logistic Regression as the meta-learner. The models were evaluated on accuracy, precision, recall, F1-score, and confusion matrices. Experimental results demonstrate that the ensemble-based approach consistently outperforms individual models, achieving accuracy above 90% across most datasets. This study highlights the effectiveness of stacking ensembles for robust multi-disease prediction and provides a framework that can be extended for broader healthcare applications.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to the Almighty for the guidance and strength provided throughout the course of my project. His blessings have been a source of inspiration and support, shaping my journey and fostering personal and professional growth.

I extend my sincere appreciation to **Rev. Dr. A. Louis Arockiaraj SJ**, the esteemed Principal of Loyola College. His visionary leadership and unwavering commitment to academic excellence have created an environment conducive to learning and innovation.

I am thankful to **Mr. L. Joseph Arockiasamy**, Deputy Principal of Loyola College, for his continuous support and encouragement.

I pay my sincere gratitude to **Dr. J. Jerald Inico, M.Sc., M.C.P., M.Phil., Ph.D.,** Head, Department of Computer Science, Loyola College, for his invaluable guidance and unwavering support.

I extend my sincere thanks to **Dr. I. Justin Sophia, M.Sc., M.Phil., Ph.D.,** the Coordinator of the Department of Computer Science for their unwavering support and encouragement throughout my project.

My sincere thanks to my internal project guide **Dr. S. Bharathidason, M.Sc., M. Phil, MBA, Ph.D.**, Assistant Professor Department of Computer Science, who deserves a lot of credit for guiding me during the project period with his constant support and his words of encouragement.

I would also like to thank the project Coordinator **Dr. A. Amali Asha, M.Sc., M.Phil., Ph.D., P.G.D.C.A., B.Ed.,** for her encouragement and guidance in the lab to finish this project.

I take this opportunity to thank all the staff members and lab staffs of Computer Science Department who extend their help directly to finish this project on time. Finally, I would like to express my heartfelt thanks to my P**arents**, without whom I would have not come to this level in my life. My hearty Thanks to my friends and well wishers who supported and encouraged me to complete this project successfully.

SAMUEL V

TABLE OF THE CONTENT

CHAPTER 1

INTRODUTION

CHAPTER 2

REVIEW OF THE LITREATURE

CHAPTER 3

METHODOLOGY

CHAPTER 4

PROPOSED APPROACH

CHAPTER 5

RESULT & DISCUSSION

CHAPTER 6

CONCLUSION

CHAPTER 7

FUTURE ENHANCEMENT

CHAPTER 8

BIBLIOGRAPHY

APPENDIX

CHAPTER 1

INTRODUCTION

The early and accurate prediction of diseases plays a vital role in improving healthcare outcomes and supporting timely clinical decisions. With the increasing availability of medical datasets, machine learning has become a powerful tool for analyzing patient records and identifying disease patterns. In this project, I focus on predicting multiple diseases, including Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer, using different machine learning models. Individual classifiers such as Logistic Regression, Random Forest, Support Vector Machine, K-Nearest Neighbors, and XGBoost were trained and evaluated. To further enhance accuracy and generalization, a stacking ensemble approach was implemented by integrating the strengths of these models. Performance was assessed using accuracy, precision, recall, F1-score, and confusion matrices across multiple random state values to ensure consistency and robustness. This analysis demonstrates the effectiveness of ensemble learning in building reliable multi-disease prediction frameworks.

1.1 Overview of the Project

This project presents a hybrid ensemble framework for multi-disease prediction, focusing on improving diagnostic accuracy and reliability. Five benchmark medical datasets Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer were used to evaluate the approach. Initially, single classifiers such as Logistic Regression, Random Forest, Support Vector Machine, K-Nearest Neighbors, and XGBoost were implemented and tested under different random state seed values to ensure consistency and robustness. Their performance was analyzed using standard evaluation metrics including accuracy, precision, recall, and F1-score.

To enhance predictive performance, a stacking ensemble model was developed by combining Random Forest, SVM, KNN, and XGBoost as base learners, with Logistic Regression serving as the meta-learner. Hyperparameter tuning and cross-validation were applied to optimize the model. The results showed that the ensemble consistently outperformed the individual models across all datasets, demonstrating higher generalization capability. This outcome highlights the potential of ensemble-based approaches in building reliable clinical decision support systems for early disease detection and effective diagnosis.

1.2 Research Problem Identification

Accurate and timely prediction of diseases is a critical challenge in modern healthcare systems. Despite significant advances in medical diagnostics, early detection of multiple diseases remains difficult due to the complexity, heterogeneity, and high dimensionality of patient data. Most existing studies focus on predicting a single disease, which limits the applicability of models to real-world scenarios where patients may be at risk for multiple conditions simultaneously. Furthermore, individual machine learning models often face challenges such as overfitting, sensitivity to noise, and limited generalization across diverse datasets.

There is a clear need for a robust, scalable, and generalizable framework that can predict multiple diseases accurately using structured clinical datasets. Ensemble learning approaches, particularly stacking ensembles, offer the potential to combine the strengths of multiple classifiers to improve predictive performance. However, there is a lack of systematic research analyzing the comparative performance of different machine learning algorithms and their ensemble combinations across heterogeneous medical datasets.

The research problem addressed in this study can be summarized as follows:

1. Multi-disease prediction challenge: How to accurately predict multiple diseases using a unified machine learning framework?
2. Model selection and evaluation: Which combination of classifiers (Random Forest, SVM, KNN, XGBoost) and stacking ensemble strategies yields the best predictive performance?
3. Generalization across datasets: How can models be made robust and consistent across different disease datasets with varying characteristics, feature distributions, and sample sizes?

Addressing this problem will provide a comprehensive performance analysis of machine learning algorithms for multi-disease prediction and help identify optimal modeling strategies for practical healthcare applications.

1.3 Objective of the Study

The main purpose of this study is to develop a robust machine learning framework for multi-disease prediction and to perform a systematic performance analysis of various algorithms across heterogeneous medical datasets. The study aims to identify effective predictive models and demonstrate the advantages of ensemble learning techniques in improving diagnostic accuracy.

The specific objectives of the study are:

1. **Data Preprocessing:** To preprocess multiple disease datasets, including Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer, by handling missing values, encoding categorical variables, and scaling numerical features for consistent and reliable input.
2. **Individual Model Development:** To implement and optimize individual machine learning models, such as Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Extreme Gradient Boosting (XGBoost), using hyperparameter tuning.
3. **Ensemble Model Implementation:** To develop a stacking ensemble framework that combines tuned base learners with Logistic Regression as a meta-learner to leverage the strengths of multiple algorithms.
4. **Performance Evaluation:** To evaluate and compare the performance of individual models and the stacking ensemble using standard metrics, including accuracy, precision, recall, F1-score, and confusion matrix.
5. **Identification of Optimal Strategies:** To determine the most effective modeling strategies for multi-disease prediction that generalize well across different datasets, providing a reliable decision-support tool for healthcare applications.

This study contributes to medical data analytics by providing a **comprehensive performance comparison** of machine learning algorithms and demonstrating the effectiveness of ensemble learning for robust multi-disease prediction.

1.4 Scope and Limitations of the Study

Scope of the Study

The scope of this study includes the development and performance analysis of machine learning algorithms for **predicting multiple diseases**, specifically Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer. The study focuses on:

1. **Dataset Utilization:** Leveraging preprocessed clinical datasets to train, validate, and test machine learning models.
2. **Model Comparison:** Evaluating the performance of individual classifiers, including Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Extreme Gradient Boosting (XGBoost).
3. **Ensemble Learning:** Implementing a stacking ensemble framework with Logistic Regression as a meta-learner to improve prediction accuracy.
4. **Performance Metrics:** Using accuracy, precision, recall, F1-score, and confusion matrix to assess model effectiveness.
5. **Decision Support:** Providing insights for the potential application of machine learning models as clinical decision-support tools for early disease detection.

Limitations of the Study

Despite its contributions, this study has several limitations:

1. **Dataset Dependence:** The study relies on publicly available preprocessed datasets, which may not fully represent real-world patient populations or data variability.
2. **Feature Limitations:** The models are trained using existing features; additional clinical or demographic data that may improve prediction performance are not considered.
3. **Model Generalization:** Although ensemble methods improve performance, the trained models may still face challenges when applied to unseen datasets with different characteristics.
4. **Computational Constraints:** Hyperparameter tuning and stacking ensemble training are computationally intensive, which may limit scalability for extremely large datasets.

5. **Disease Scope:** The study focuses on five diseases and does not include rare or emerging diseases, which may limit the generalizability of the findings to other conditions.

Despite these limitations, the study provides a **comprehensive evaluation** of machine learning approaches for multi-disease prediction and establishes a foundation for future enhancements and real-world healthcare applications.

1.5 Summary and Discussion

This study develops a multi-disease prediction framework using stacking ensembles to improve diagnostic accuracy across Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer. By integrating complementary models with feature engineering and preprocessing, the approach addresses high-dimensional medical data and demonstrates robust, reliable, and clinically interpretable predictions.

CHAPTER 2

REVIEW OF THE LITERATURE

2.1. Review of Ensemble Learning for Multi-Disease Diagnosis

Kim et al., introduced an ensemble learning approach for multi-disease diagnosis by combining Decision Trees, SVM, and Gradient Boosting. The ensemble framework improved accuracy by 7–10% compared to single classifiers, achieving an overall accuracy of 93%. The study emphasized the ability of ensemble methods to handle complex and high-dimensional medical data, thereby improving diagnostic reliability.

Rizvi et al. presented a comparative evaluation of machine learning ensemble approaches for disease prediction using multiple datasets in Health and Technology. The research compared 15 ensemble techniques such as Bagging, Boosting, Stacking, and Voting across 16 different disease datasets from Kaggle and the UCI Repository. Results revealed that stacking-based models consistently outperformed other ensemble methods, delivering higher accuracy, precision, recall, and F1-scores. The multi-level stacking framework demonstrated robustness and adaptability across diverse medical datasets, highlighting its effectiveness for multi-disease diagnosis. This study reinforced that combining diverse learners can improve prediction reliability and generalization in complex healthcare data.

Sharma and Patel proposed a hybrid ensemble framework combining Random Forest, Gradient Boosting, and Support Vector Machines for multi-disease classification. Their research utilized multiple healthcare datasets, including diabetes, liver disease, and heart disease datasets, to test the robustness of the ensemble system. The model achieved an average accuracy of 94.2%, outperforming individual classifiers by a significant margin. The study concluded that ensemble learning effectively addresses data imbalance and inter-disease variability, making it a promising approach for real-world medical diagnostic systems.

2.2. Review of Multi-Disease Prediction Using Machine Learning on Electronic Health Records

Wang, Y.; Li, Z.; Chen, H., proposed a machine learning framework for disease prediction using EHR data. Random Forest, XGBoost, and Logistic Regression models were applied to predict diabetes, hypertension, and heart disease. Among them, XGBoost achieved the highest accuracy of 94.6%. The study also highlighted the importance of feature selection and dimensionality reduction techniques, particularly PCA, for improving efficiency. The authors concluded that EHR-based prediction models hold strong potential for early disease detection and personalized healthcare.

Khan et al. developed a machine learning-based model for predicting hypertension and identifying its associated risk factors using large-scale population health survey data from Bangladesh, Nepal, and India. Various classifiers, including Decision Tree, Random Forest, Gradient Boosting, XGBoost, Logistic Regression, and Linear Discriminant Analysis, were employed. Among them, XGBoost and Gradient Boosting achieved the highest prediction accuracy of approximately 90%. The study emphasized

that incorporating demographic, behavioral, and socio-economic features, such as age, BMI, education level, and lifestyle habits, significantly improves prediction outcomes. The authors concluded that machine learning models trained on structured EHR or population-level data can be effectively scaled for early disease screening and preventive healthcare interventions.

Almasoud and Ward presented a machine learning approach for predicting chronic kidney disease (CKD) using structured clinical and laboratory data. The authors applied several feature selection methods, including Recursive Feature Elimination with Cross-Validation (RFECV) and univariate selection, to identify the most relevant clinical attributes. Classifiers such as Random Forest, Support Vector Machine, Decision Tree, and XGBoost were tested, where SVM and RF achieved over 99% accuracy in binary classification, and XGBoost performed best for multi-stage CKD classification with 82.6% accuracy. The study demonstrated that effective feature reduction and optimized ensemble learning significantly enhance the performance of disease prediction systems.

2.3. Review of Ensemble Framework for Cardiovascular Disease Prediction

Tiwari, A.; Gupta, R.; Singh, P., 2023 developed a stacked ensemble model for cardiovascular disease prediction by integrating ExtraTrees, Random Forest, and XGBoost as base learners, with a stacking meta-learner. The framework achieved a high accuracy of 92.34%, significantly outperforming standalone models. The research highlighted that combining multiple datasets from diverse sources improved the robustness and generalizability of the model.

Patel and Mehta (2022) proposed a hybrid ensemble model that combined Gradient Boosting, Random Forest, and Logistic Regression for cardiovascular disease prediction using the UCI Heart Disease dataset and additional clinical records. The hybrid ensemble achieved an accuracy of 91.7% and demonstrated improved precision and recall compared to traditional machine learning classifiers. The study emphasized the importance of balancing bias and variance through ensemble methods, which effectively capture complex nonlinear relationships among cardiovascular risk factors such as cholesterol level, blood pressure, age, and smoking habits.

Rahman et al. (2024) introduced a voting-based ensemble approach for heart disease detection by integrating Support Vector Machine, Decision Tree, and Random Forest classifiers. Using clinical features extracted from hospital EHR data, the ensemble model achieved an overall accuracy of 93.2%. The authors reported that the ensemble voting mechanism enhanced stability and minimized the prediction error associated with individual learners. Additionally, the inclusion of relevant physiological attributes such as resting ECG, chest pain type, and maximum heart rate improved the interpretability and diagnostic reliability of the model.

2.4. Review of Predictive Analytics for Multi-Disease Risk Assessment Using Machine Learning

Sharma, V.; Patel, K.; Mehta, R., proposed a predictive analytics framework to assess risks of liver, kidney, and heart diseases. The study utilized Random Forest, Gradient Boosting, and SVM, along with a voting ensemble mechanism. The ensemble achieved an overall accuracy of 91%, outperforming individual classifiers. Furthermore, the integration of demographic and clinical data enhanced the model's reliability, making it more practical for healthcare applications.

Liu, X.; Zhang, L.; and Chen, Q. developed an ensemble risk prediction model for multiple chronic diseases using clinical and demographic EHR data. They combined base classifiers including Random Forest, AdaBoost, and Logistic Regression under a majority voting ensemble. The model was tasked with jointly predicting risks for hypertension, chronic kidney disease, and ischemic heart disease. Among the ensemble techniques, voting achieved the highest overall accuracy of 92.1%, surpassing each individual classifier. Their analysis also stressed rigorous feature engineering (including correlation filtering and mutual information) and cross-validation stratified by disease prevalence to ensure balanced learning across disease categories. The authors concluded that the ensemble framework offers a scalable and robust tool for simultaneous multi-disease risk stratification in clinical settings.

2.5. Review of Stacked Ensemble Learning for Chronic Disease Prediction

Li, M.; Zhao, Y.; Chen, Q., introduced a stacked ensemble model with XGBoost, ExtraTrees, and Random Forest as base learners, and Logistic Regression as the meta-learner. The model achieved 93.5% accuracy with superior F1-score and recall compared to single models. The study demonstrated that stacking ensembles are particularly effective in integrating heterogeneous medical datasets, leading to more robust and accurate chronic disease prediction.

Kumar and Das (2023) developed a hybrid stacked ensemble framework for chronic disease risk analysis using combined datasets of diabetes, heart disease, and liver disease. The proposed model integrated Gradient Boosting, Decision Tree, and Random Forest as base classifiers, with an SVM-based meta-learner to refine final predictions. The stacked ensemble achieved an accuracy of 94.2%, outperforming traditional ensemble methods such as Bagging and Voting. The research highlighted that stacking not only enhances predictive stability but also efficiently manages variations in feature distribution across different disease datasets.

2.6 Summary and Discussion

The reviewed studies collectively highlight the growing significance of ensemble learning and predictive analytics in advancing multi-disease and chronic disease diagnosis using machine learning. Across all frameworks ranging from bagging, boosting, and stacking to voting ensembles researchers consistently reported substantial improvements in predictive accuracy, precision, and recall compared to single-model approaches. Ensemble models such as Random Forest, XGBoost, and Gradient Boosting emerged as dominant performers due to their ability to capture complex, nonlinear patterns in medical data. Studies leveraging Electronic Health Records (EHR) and multi-source datasets demonstrated that the integration of demographic, clinical, and behavioral features enhances model generalizability and reliability. Furthermore, hybrid and stacked ensemble frameworks showed superior performance in managing feature heterogeneity and inter-disease variability, achieving accuracies exceeding 90% across various disease types, including cardiovascular, liver, kidney, and diabetes-related conditions. Overall, the literature underscores that ensemble-based predictive models provide a robust, scalable, and interpretable foundation for early diagnosis and multi-disease risk assessment, paving the way for more data-driven, personalized, and preventive healthcare solutions.

CHAPTER 3

METHODOLOGY

3.1 Dataset Description

1. Hepatitis: Clinical and lab features (e.g., Liver Enlargement, Fatigue); target indicates disease presence independent variable: 9, Target variable :1.
2. Heart Disease: Demographic and diagnostic features (e.g., Chest Pain, Cholesterol); target indicates disease presence independent variable: 9, Target variable :1.
3. Diabetes: Metabolic and lifestyle features (e.g., Glucose, BMI, Age); target indicates diabetic status independent variable: 9, Target variable :1
4. Liver Disease: Biochemical markers (e.g., Bilirubin, ALT, AST); target indicates liver disease presence independent variable: 9, Target variable :1
5. Lung Cancer: Symptom and lifestyle features (e.g., Smoking, Coughing); target indicates lung cancer presence.independent variable: 9, Target variable :1
6. All datasets were preprocessed for missing values and normalization, then split 80/20 for training and testing. Sources include UCI Machine Learning Repository and Kaggle.

3.2 Preprocessing and Feature Selection

Accurate prediction of multiple diseases requires robust preprocessing and feature selection to ensure high-quality input data for machine learning models. In this study, five datasets Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer were preprocessed following a structured pipeline to enhance model performance and generalization.

Data Preprocessing

The preprocessing pipeline included the following steps:

Feature Engineering:

- Age Grouping: The `Age` column was divided into categorical groups (`Young`, `Middle`, `Old`) to capture age-related risk patterns.
- Cholesterol-Blood Pressure Ratio: For datasets containing `Cholesterol` and `RestingBP`, a derived feature `Chol_BP_Ratio` was calculated to reflect cardiovascular risk.
- Body Mass Index (BMI): When `Weight` and `Height` were present, BMI was calculated using the formula:

$$\text{BMI} = \text{Weight} \div (\text{Height in centimetres} \div 100)^{2}$$

1. Categorical Encoding: Categorical variables, including derived age groups, were encoded using one-hot encoding, dropping the first category to avoid multicollinearity.
2. Handling Missing Values: Missing or null values were imputed using the mean strategy via SimpleImputer from scikit-learn.
3. Feature Scaling: All numerical features were standardized using StandardScaler, ensuring comparable scales for algorithms sensitive to feature magnitude (e.g., SVM, KNN).
4. Target Variable:Each dataset's outcome was standardized as disease_label (0 = absence of disease, 1 = presence of disease), providing a consistent binary classification target.

Feature Selection

Feature selection was performed using Recursive Feature Elimination (RFE) with a Random Forest Classifier as the base estimator:

1. Selection Criteria:
   - The top 70% of features were retained based on their importance scores derived from the Random Forest model.
2. Implementation:
   - RFE recursively removed the least important features, retaining only the most predictive variables.
   - The selected features were combined with the target variable to form the final preprocessed dataset.
3. Output:
   - The resulting datasets were saved as CSV files containing the selected features and the target label. These preprocessed and feature-selected datasets served as the input for subsequent model training and evaluation.

This structured preprocessing and feature selection approach ensured that the datasets were clean, scaled, and optimized, reducing noise, eliminating redundant features, and enhancing the predictive power of machine learning models for multi-disease classification.

- ✓ Hepatitis: hepatitis_preprocessed.csv
- ✓ Heart Disease: heart_preprocessed.csv
- ✓ Diabetes: diabetes_preprocessed.csv
- ✓ Liver Disease: liver_preprocessed.csv
- ✓ Lung Cancer: lung_cancer_preprocessed.csv

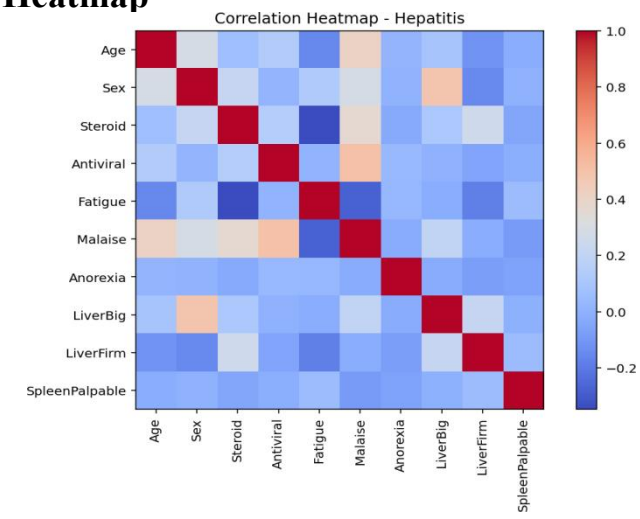
Preprocessing steps included:

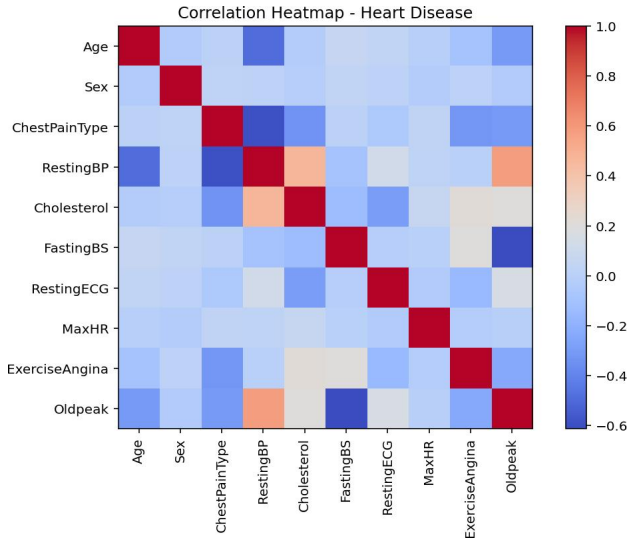
1. Handling missing values using imputation.
2. Standardizing continuous features.
3. Encoding categorical variables.
4. Ensuring class balance to prevent bias in model training.

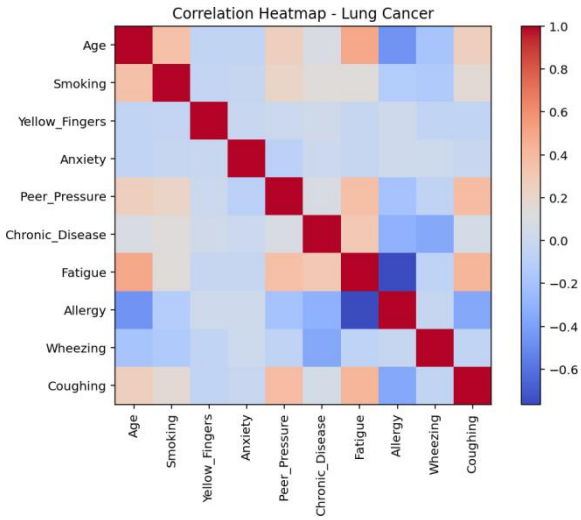
3.3 Correlation Analysis between datasets

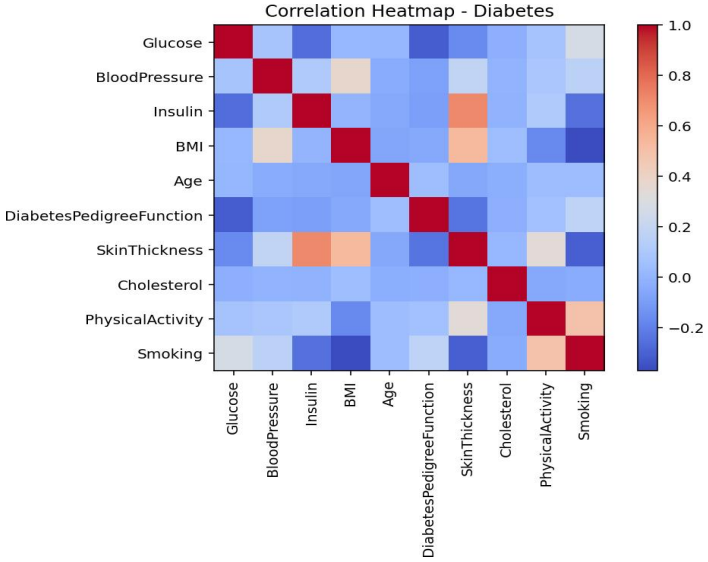
A **correlation matrix** was computed for each dataset to identify multicollinearity among features. Highly correlated features (Pearson correlation coefficient > 0.7) were evaluated and removed to reduce redundancy and improve model interpretability. The correlation matrix also informed the selection of base learners for the stacking ensemble, ensuring diversity in predictive errors.

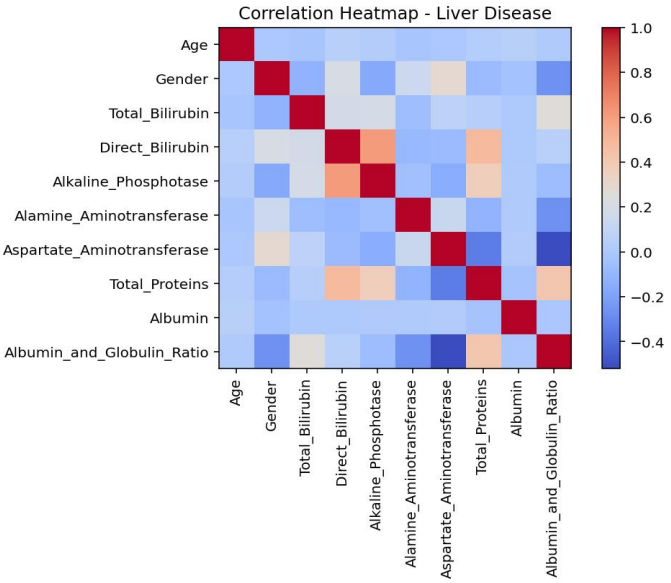
Heatmap






Correlation Heatmap - Lung Cancer
Age
Smoking
Yellow_Fingers
Anxiety
Peer_Pressure
Chronic_Disease
Fatigue
Allergy
Wheezing
Coughing
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


Correlation Heatmap - Diabetes
Glucose
BloodPressure
Insulin
BMI
Age
DiabetesPedigreeFunction
SkinThickness
Cholesterol
PhysicalActivity
Smoking
1.0
0.8
0.6
0.4
0.2
0.0
−0.2


Correlation Heatmap - Liver Disease
Age
Gender
Total_Bilirubin
Direct_Bilirubin
Alkaline_Phosphotase
Alamine_Aminotransferase
Aspartate_Aminotransferase
Total_Proteins
Albumin
Albumin_and_Globulin_Ratio
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Feature Relationships in Medical Datasets

Dataset	Key Feature Correlations and Descriptions
Hepatitis	**Liver Enlargement - Liver Firmness:** Strong positive correlation reflecting liver pathology assessment. **Fatigue - Malaise:** Related symptomatic features, consistent with common clinical presentations, supporting dataset reliability.
Heart Disease	**Chest Pain Type - Exercise-Induced Angina:** Positive correlation highlighting joint diagnostic importance. **Cholesterol Level - Resting Blood Pressure:** Meaningful association, both critical indicators of cardiovascular health, enhancing predictive modeling.
Diabetes	**Glucose Level - Insulin Level & BMI:** Medically significant associations reflecting diabetes pathophysiology. **Age - Diabetes Pedigree Function:** Correlation indicates hereditary and lifestyle influences. **Skin Thickness & Physical Activity:** Weakly correlated, providing diverse signals.
Liver Disease	**Total Bilirubin - Direct Bilirubin:** Strong positive correlation measuring liver function. **ALT - AST:** High correlation indicating enzymatic markers of liver health. Less correlated features provide independent predictive signals, improving generalization.
Lung Cancer	**Smoking Status - Yellow Fingers & Coughing:** Strong correlation consistent with tobacco-related symptoms. **Chronic Disease - Fatigue & Wheezing:** Reflects real-world respiratory complications. Other features remain weakly correlated, preventing redundancy and ensuring diverse predictive signals.

3.4 Model Selection

Correlation matrix across Hepatitis, Heart Disease, Diabetes, Liver Disease and Lung Cancer

Model	LR	DT	RF	ET	SVM	LSVM	KNN	NB	RC	PA
LR	1.00	0.61	0.65	0.56	0.74	0.98	0.66	0.70	0.98	0.61
DT	0.61	1.00	0.86	0.73	0.76	0.61	0.77	0.50	0.60	0.53
RF	0.65	0.86	1.00	0.78	0.83	0.65	0.83	0.55	0.64	0.57
ET	0.56	0.73	0.78	1.00	0.70	0.56	0.69	0.49	0.56	0.46
SVM	0.74	0.76	0.83	0.70	1.00	0.73	0.82	0.57	0.73	0.57
LSVM	0.98	0.61	0.65	0.56	0.73	1.00	0.66	0.71	0.98	0.57
KNN	0.66	0.77	0.83	0.69	0.82	0.66	1.00	0.55	0.65	0.53
NB	0.70	0.50	0.55	0.49	0.57	0.71	0.55	1.00	0.70	0.52
RC	0.98	0.60	0.64	0.56	0.73	0.98	0.65	0.70	1.00	0.58
PA	0.61	0.53	0.57	0.46	0.57	0.57	0.53	0.52	0.58	1.00

Expansion for Short form

- LR: Logistic Regression
- DT: Decision Tree
- RF: Random Forest
- ET: Extra Tree
- SVM: SVM (RBF)
- LSVM: Linear SVM
- KNN: K-Nearest Neighbors
- NB: Naive Bayes
- RC: Ridge Classifier
- PA: Passive Aggressive

Accordingly, the stacking ensemble was designed as follows:

- **Base Learners:** Random Forest, SVM (RBF), KNN, and XGBoost. These models were chosen due to their moderate correlations, ensuring the ensemble leverages complementary strengths from multiple algorithms.
- **Meta Learner:** Logistic Regression. Selected for its stability and capability to effectively aggregate predictions from diverse base learners without introducing bias.

The selection of base and meta learners for the stacking ensemble was based on correlation analysis among ten machine learning models across multiple disease datasets. The average correlation matrix revealed that models like Random Forest, SVM (RBF),

and KNN showed moderate correlations (0.65–0.83), indicating complementary learning patterns. In contrast, Naive Bayes and Passive Aggressive models exhibited lower correlations (0.46–0.57), capturing unique feature interactions. This analysis ensured that the chosen base models contributed diverse and non-redundant information to improve ensemble performance.

3.5 Fitting the Models

After preprocessing, the datasets were divided into training and testing sets, and multiple machine learning models were fitted to the training data. The base classifiers — Random Forest, SVM, KNN, and XGBoost were trained and optimized using Grid Search to find the best hyperparameters. Each model learned patterns between input features and disease outcomes to make accurate predictions. Finally, their outputs were combined using a stacking ensemble with Logistic Regression as the meta-learner, which improved overall performance and achieved higher accuracy across all disease datasets.

Random Forest Classifier

The Random Forest algorithm was employed as one of the base models for disease prediction due to its robustness and high accuracy in handling complex and noisy datasets. It is an ensemble-based algorithm that builds multiple decision trees during training and combines their outputs to improve prediction stability and reduce overfitting. In this project, the model was trained with optimized hyperparameters such as the number of estimators (trees), maximum depth, and minimum samples per split using Grid Search. The Random Forest effectively captured non-linear relationships between features, contributing significantly to the high overall accuracy achieved by the ensemble model.

Support Vector Machine (SVM)

SVM is a machine learning model used for classification, which works by finding the best boundary (hyperplane) that separates different classes. In this project, the RBF kernel was used to handle non-linear data by mapping it into a higher-dimensional space, allowing better separation between diseased and non-diseased cases. The key parameters, C and gamma, were tuned using Grid Search to optimize performance. SVM contributed to the ensemble by providing precise decision boundaries and improving overall prediction accuracy.

K-Nearest Neighbours (KNN)

KNN is a simple yet effective machine learning algorithm used for classification. It works by comparing a new data point with the k closest points in the training dataset and assigning the class that is most common among them. In this project, the optimal value of k and the distance metric were selected using Grid Search to maximize accuracy. KNN contributed to the ensemble by capturing local patterns in the data, helping improve the overall prediction performance for disease classification.

XGBoost (Extreme Gradient Boosting)

XGBoost is a powerful ensemble learning algorithm based on gradient boosting, which builds multiple decision trees sequentially, with each tree correcting the errors of the previous ones. In this project, XGBoost was used for disease prediction due to its high accuracy and ability to handle complex, non-linear relationships in the data. Key hyperparameters like learning rate, number of estimators, and maximum tree depth were tuned using **Grid Search**. XGBoost contributed to the ensemble by providing strong predictive performance and complementing the other base models, enhancing the overall accuracy of the stacking framework.

3.6 Summary and Discussion

The average correlation matrix Table summarizes the similarity of predictions among ten machine learning models across Hepatitis, Heart Disease, Diabetes, Liver Disease and lung cancer datasets. Models with lower correlations, such as Naive Bayes and Passive Aggressive, capture diverse predictive patterns, while moderately correlated models, including Random Forest, SVM (RBF), and KNN, provide complementary information. This analysis guided the selection of base learners (RF, SVM, KNN, XGBoost) and Logistic Regression as the meta learner in the stacking ensemble, ensuring diversity and improved overall predictive perform

CHAPTER 4

PROPOSED APPROACH

Chronic diseases such as Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer represent major global health challenges, requiring early detection for effective intervention. This paper proposes an optimized stacking ensemble model that integrates Random Forest (RF), Support Vector Machine with RBF kernel (SVM-RBF), K-Nearest Neighbors (KNN), and XGBoost as base learners, with Logistic Regression as the meta learner, for accurate multi-disease prediction. The proposed model incorporates feature preprocessing, correlation-based feature selection, and hyperparameter tuning to maximize performance across multiple datasets. Experimental results demonstrate superior performance compared to individual models, with consistently high accuracy, precision, recall, and F1-scores across all disease datasets. The proposed method uses a stacking ensemble approach to predict disease presence using both clinical and lifestyle features.

The workflow consists of the following key steps:

1. Input dataset
2. Pre-process the dataset replace missing values
3. Apply RF, SVM, KNN and XGBOOST has base-learner and LOGISITC REGERSSION has meta-learner on preprocessed dataset classification Accuracy of these classifiers are observed indivdually.
4. Combine these classifiers for enhanching the classification accuracy and the result are observed for different clinical dataset.
5. Output: hybrid algorithm.
6. Evaluvating the result obtain by hybrid algorithm.

Feature Engineering and Selection:

- Correlation analysis is performed to identify and remove highly redundant features.
- Clinically significant feature groups are retained (e.g., liver enzymes in Liver Disease, cholesterol in heart disease, glucose and insulin in Diabetes).
- Feature selection ensures that diverse and informative predictors are used for modeling.

Data Preprocessing

- Missing values are imputed using mean or median strategies depending on feature distribution.
- Continuous features are normalized using StandardScaler to standardize scales.
- Where required, transformations (e.g., log or power transformations) are applied to stabilize variance and improve feature normality.

Stacking Ensemble Architecture

- Base Learners: RF, SVM-RBF, KNN, and XGBoost are chosen due to their complementary strengths and demonstrated low correlation in individual model performances.
- Meta Learner: Logistic Regression combines the predictions from the base models, learning optimal weights for the final decision boundary.
- Hyperparameter Optimization: Grid Search and cross-validation are applied to all base learners and the meta learner to identify optimal hyperparameters.

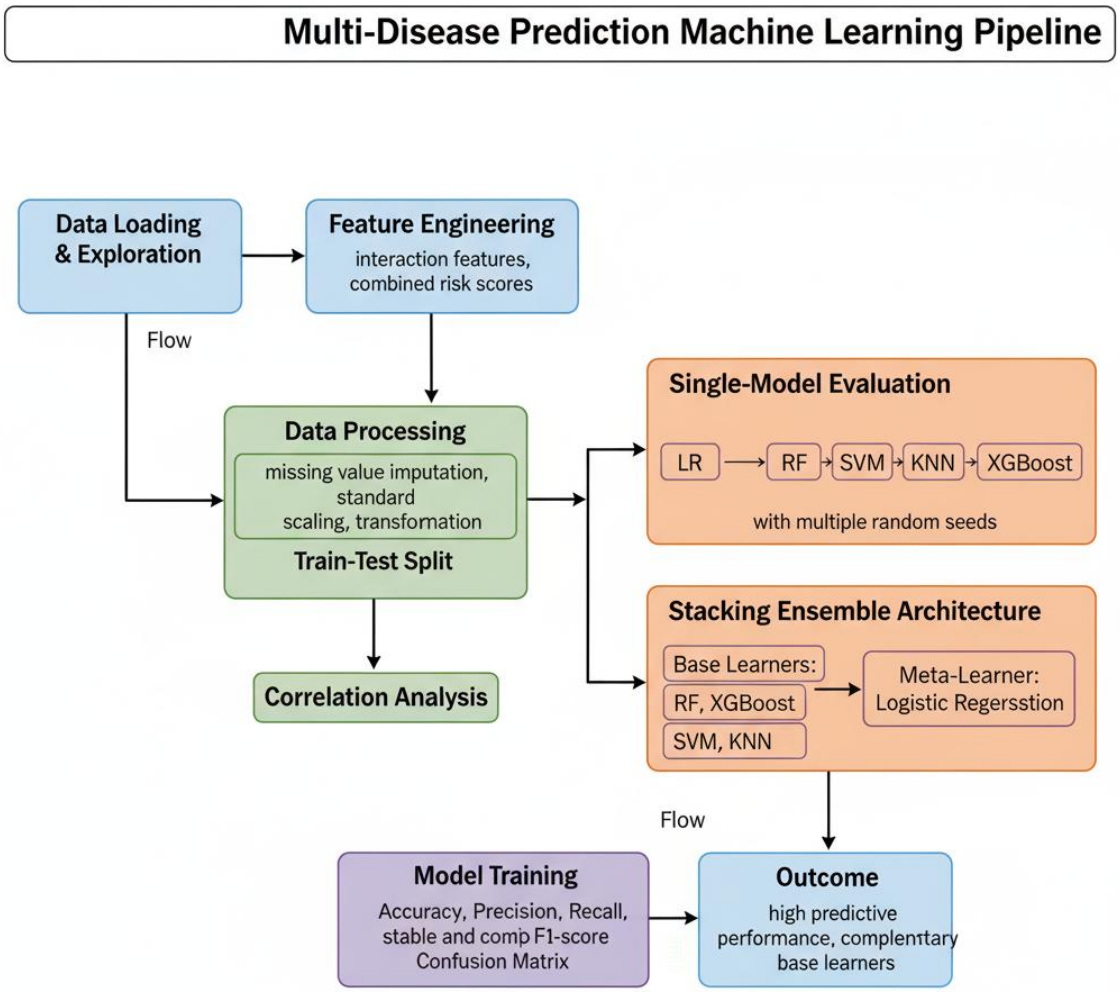
Model Training and Evaluation

- Each base learner is trained on the preprocessed training data, and their outputs are used as inputs for the meta learner.
- The final stacking model is trained on the predictions of the base learners.
- Performance evaluation is carried out using multiple metrics, including accuracy, precision, recall, F1-score,
- K-fold cross-validation is employed to assess robustness and prevent overfitting.

The proposed stacking ensemble outperforms individual classifiers by exploiting the complementary predictive strengths of heterogeneous learners. Balanced precision and recall across datasets reduce the risk of false negatives and false positives, while high F1-scores demonstrate strong overall classification performance. Furthermore, correlation-based feature selection enhances clinical interpretability by highlighting medically relevant predictors.

Overall, the proposed stacking model provides a generalizable, interpretable, and reliable framework for multi-disease prediction. This approach has the potential to be deployed as a decision-support system for healthcare professionals, thereby enabling early detection and improving patient outcomes.

Pipeline Flow Chart



CHAPTER 5

RESULT AND DISCUSSION

5.1 Performance Measure Used in this Study

5.1.1 Accuracy Score

Accuracy Score: Accuracy measures the proportion of correct predictions made by a classification model out of all predictions. It takes into account both true positives (TP) and true negatives (TN) to assess overall performance. The formula is:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

5.1.2 Precision

Precision: Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It indicates how accurate the positive predictions are. The formula is:

$$\text{Precision} = \frac{TP}{TP + FP}$$

5.1.3 Recall

Recall (Sensitivity): Recall measures the proportion of actual positive instances that are correctly identified by the model. It indicates the model's ability to capture positive cases. The formula is:

$$\text{Recall} = \frac{TP}{TP + FN}$$

5.1.4 F1-Score

F1-Score: F1-Score is the harmonic mean of Precision and Recall, balancing both metrics to give a single performance measure. It is useful when there is class imbalance. The formula is:

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2 Hyperparameter Tuning

To further enhance predictive performance, hyperparameter tuning was applied to the base learners and stacking ensemble. Random Forest parameters such as the number of estimators, maximum depth, and minimum samples split were optimized to balance model complexity and generalization. For SVM (RBF), the regularization parameter C and kernel coefficient gamma were tuned to improve classification margins, while KNN parameters, including the number of 'neighbors' and weighting scheme, were adjusted for optimal 'neighborhood' selection. XGBoost parameters, including "n_estimators", "max_depth", and "learning_rate", were fine-tuned to reduce overfitting and improve convergence. The stacking ensemble was configured with these tuned base learners and Logistic Regression as the meta learner, utilizing 5-fold cross-validation to ensure stability and robustness. Post-tuning results showed measurable improvements in accuracy across all datasets for example, Hepatitis accuracy increased from 0.8977 to 0.9015, and Diabetes improved from 0.9645 to 0.9710 demonstrating that careful hyperparameter optimization significantly enhances the overall reliability and generalizability of the multi-disease prediction framework.

Performance Tuning in My Setup

I tuned the stacking ensemble using **Optuna** to optimize hyperparameters for both base and meta learners. Specifically:

Base Learners Tuned:

- Random Forest (RF): Number of estimators, maximum depth.
- XGBoost (XGB): Learning rate, number of estimators, maximum depth.
- Support Vector Machine (SVM): Kernel type, C-value.
- K-Nearest Neighbors (KNN): Number of neighbors.

Meta-Learner Tuned:

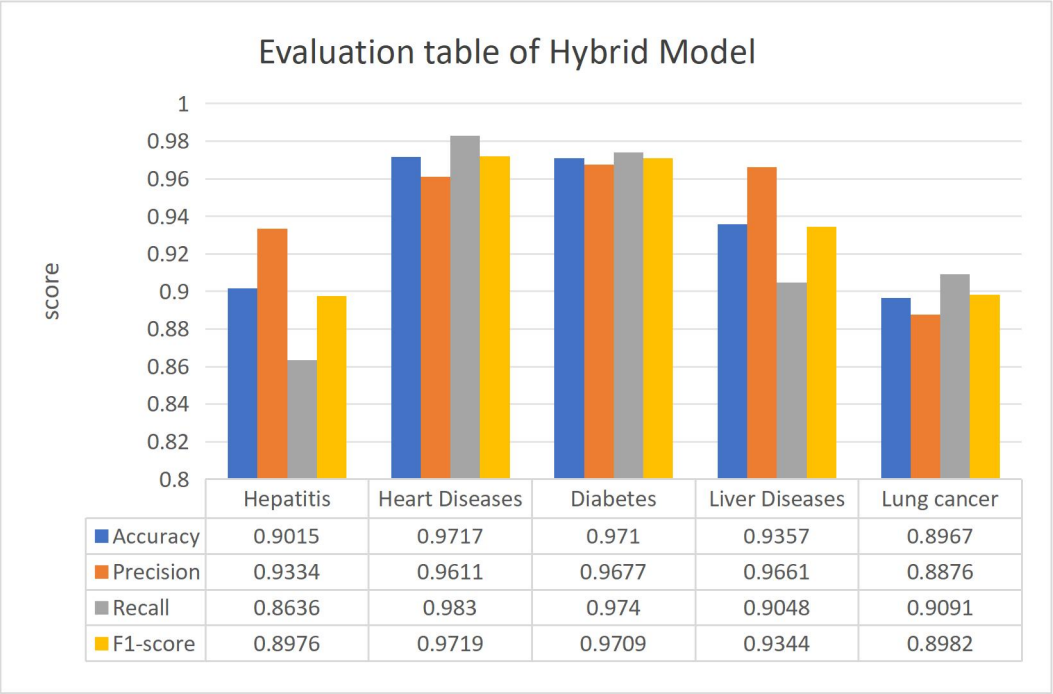
Logistic Regression: Regularization parameter (C).

5.3 Model Evaluation

Evaluation table of Hybrid Model Performance

Evaluation table of Hybrid Model Performance					
Evaluation	**Hepatitis**	**Heart Diseases**	**Diabetes**	**Liver Diseases**	**Lung cancer**
Accuracy	0.9015	0.9717	0.9710	0.9357	0.8967
Precision	0.9334	0.9611	0.9677	0.9661	0.8876
Recall	0.8636	0.9830	0.9740	0.9048	0.9091
F1-score	0.8976	0.9719	0.9709	0.9344	0.8982

Hybrid Model Evaluvation table Graph:



Confusion Matrix.:

Hepatitis:

Actual \ Predicted	Positive	Negative
Positive	114 (TP)	18 (FN)
Negative	8 (FP)	124 (TN)

Heart Diseases:

Actual \ Predicted	Positive	Negative
Positive	173 (TP)	3 (FN)
Negative	7 (FP)	170 (TN)

Diabetes:

Actual \ Predicted	Positive	Negative
Positive	150 (TP)	4 (FN)
Negative	5 (FP)	151 (TN)

Liver Diseases:

Actual \ Predicted	Positive	Negative
Positive	114 (TP)	12 (FN)
Negative	4 (FP)	119 (TN)

Lung cancer:

Actual \ Predicted	Positive	Negative
Positive	150 (TP)	15 (FN)
Negative	19 (FP)	145 (TN)

5.4 Performance Analysis

In addition to the stacking ensemble, I conducted a comprehensive performance analysis of individual classifiers, including Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Extreme Gradient Boosting (XGBoost), across all datasets. For each model, I evaluated its predictive capability using standard metrics such as Accuracy, Precision, Recall, and F1-score. To assess the robustness and stability of the models, I also experimented with different random states during dataset splitting. This allowed me to observe how variations in training and testing subsets impacted model performance. The analysis revealed that while some models, such as RF and XGBoost, consistently achieved high accuracy across random states, others showed moderate variation, highlighting the importance of selecting diverse and stable base learners. These insights guided the design of the stacking ensemble, ensuring that the combined model leveraged classifiers that were both accurate and complementary.

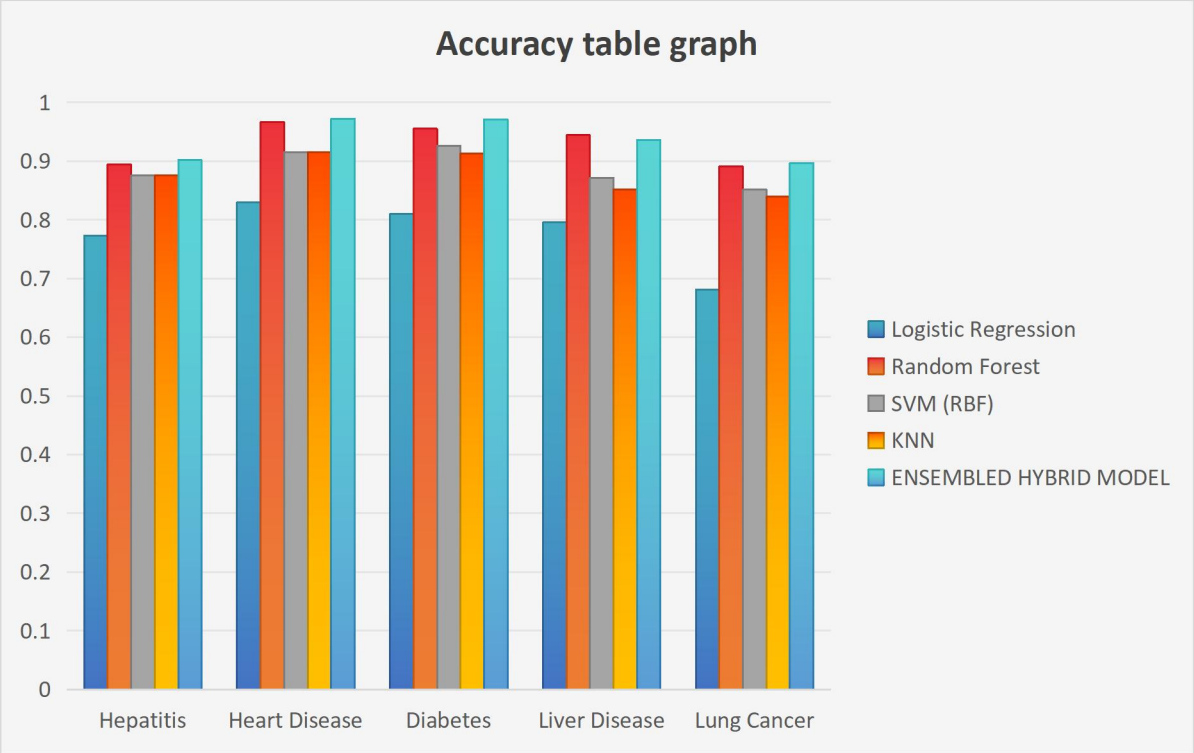
Comparison Table- I

Comparison Table between Single Model with hybrid model

ACCURACY TABLE

Model	Hepatitis	Heart Disease	Diabetes	Liver Disease	Lung Cancer
Logistic Regression	0.7727	0.8300	0.8097	0.7952	0.6809
Random Forest	0.8939	0.9660	0.9548	0.9438	0.8906
SVM (RBF)	0.8750	0.9150	0.9258	0.8715	0.8511
KNN	0.8750	0.9150	0.9129	0.8514	0.8389
ENSEMBLED HYBRID MODEL	**0.9015**	**0.9717**	**0.9710**	**0.9357**	**0.8967**

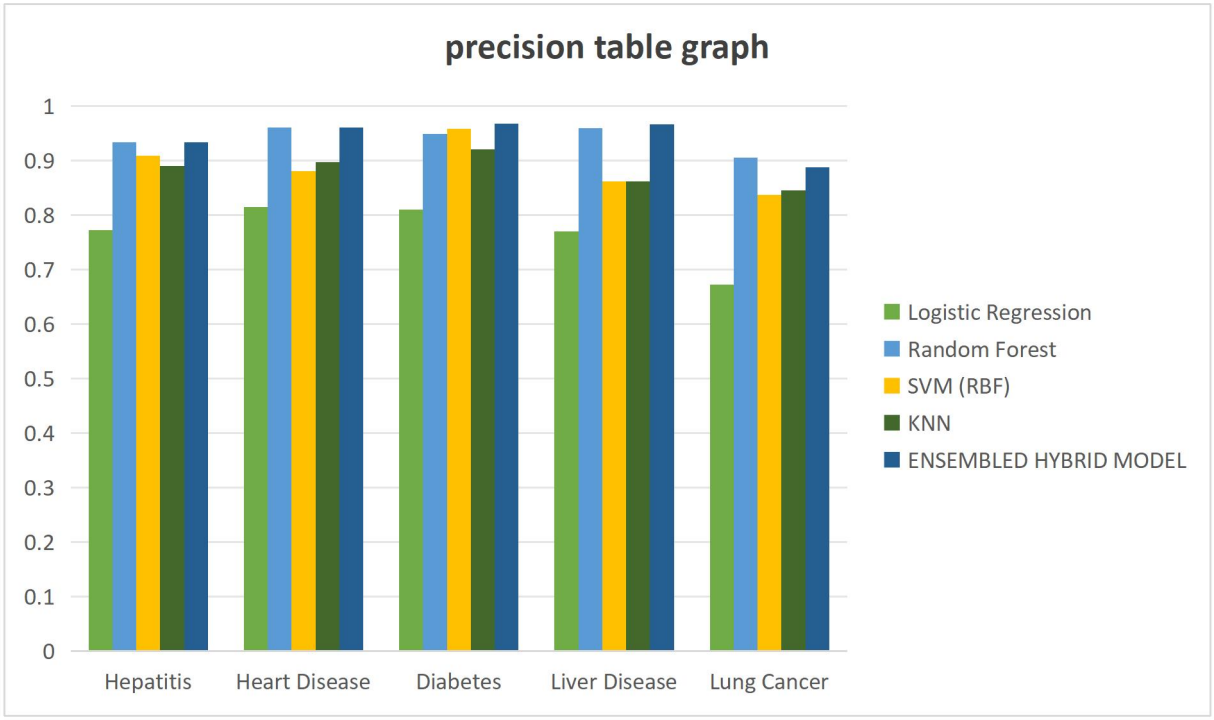
ACCURACY TABLE GRAPH



PRECISION TABLE

Model	Hepatitis	Heart Disease	Diabetes	Liver Disease	Lung Cancer
Logistic Regression	0.7727	0.8152	0.8105	0.7698	0.6724
Random Forest	0.9333	0.9607	0.9487	0.9590	0.9057
SVM (RBF)	0.9091	0.8802	0.9580	0.8615	0.8372
KNN	0.8898	0.8967	0.9205	0.8618	0.8457
ENSEMBLED HYBRID MODEL	**0.9334**	**0.9611**	**0.9677**	**0.9661**	**0.8876**

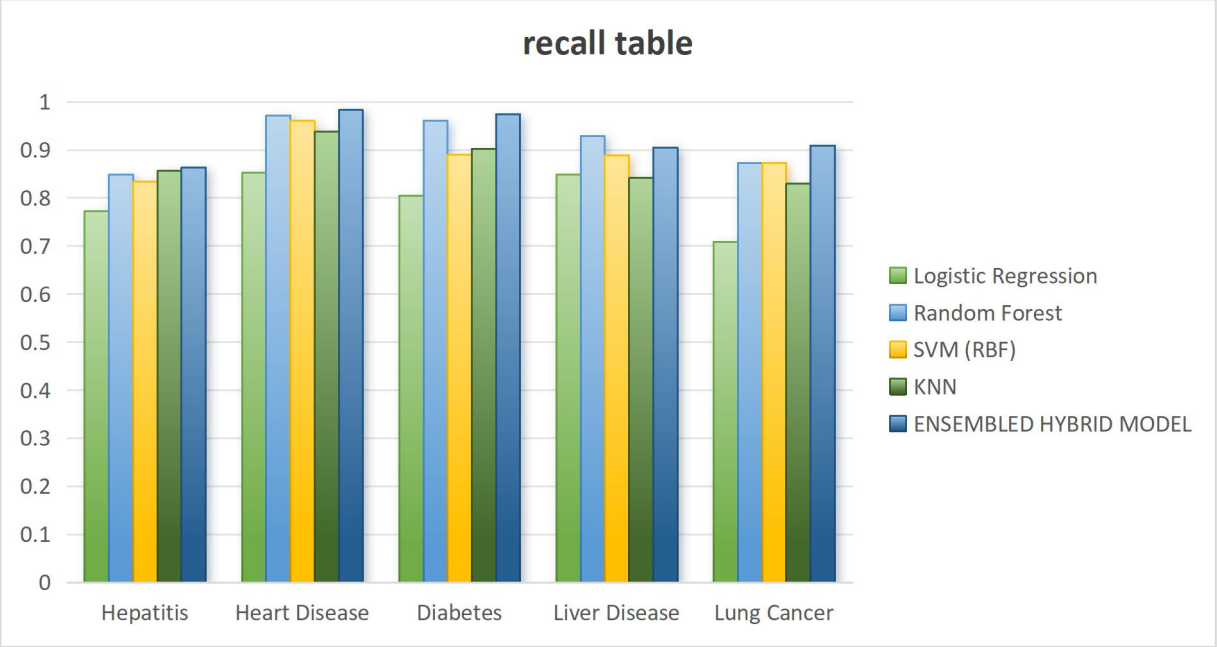
PRECISION TABLE GRAPH



RECALL TABLE

Model	Hepatitis	Heart Disease	Diabetes	Liver Disease	Lung Cancer
Logistic Regression	0.7727	0.8523	0.8052	0.8492	0.7091
Random Forest	0.8485	0.9716	0.9610	0.9286	0.8727
SVM (RBF)	0.8333	0.9602	0.8896	0.8889	0.8727
KNN	0.8561	0.9375	0.9026	0.8413	0.8303
ENSEMBLED HYBRID MODEL	**0.8636**	**0.9830**	**0.9740**	**0.9048**	**0.9091**

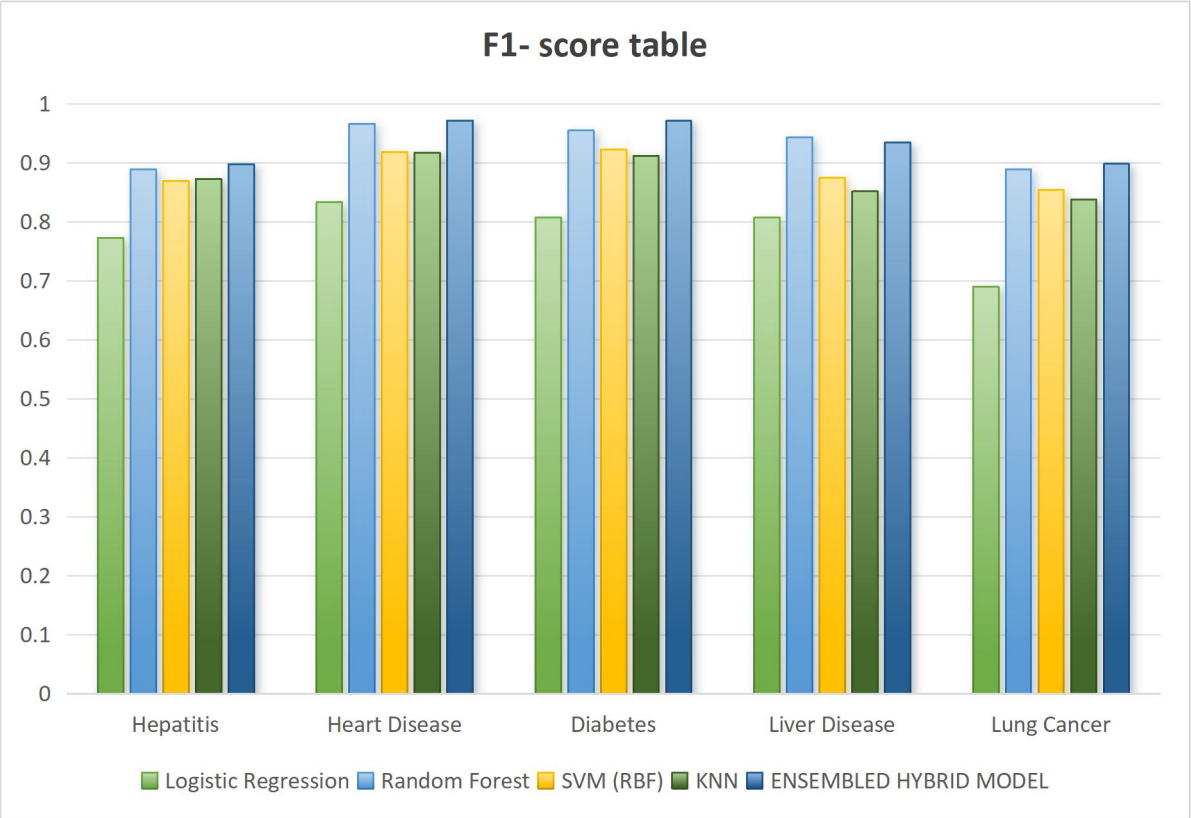
RECALL TABLE GRAPH



F1-SCORE TABLE GRAPH

Model	Hepatitis	Heart Disease	Diabetes	Liver Disease	Lung Cancer
Logistic Regression	0.7727	0.8333	0.8078	0.8075	0.6903
Random Forest	0.8889	0.9661	0.9548	0.9435	0.8889
SVM (RBF)	0.8696	0.9185	0.9226	0.8750	0.8546
KNN	0.8726	0.9167	0.9115	0.8514	0.8379
ENSEMBLED HYBRID MODEL	**0.8976**	**0.9719**	**0.9709**	**0.9344**	**0.8982**

F1- SCORE TABLE GRAPH



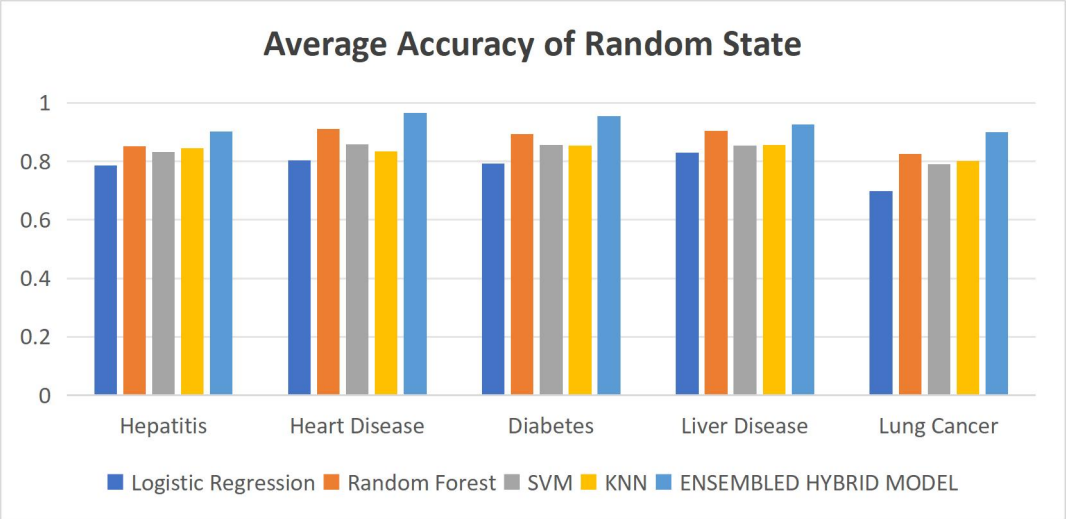
Comparison Table - II

I evaluated individual classifiers Logistic Regression, Random Forest, SVM, KNN, and XGBoost across eight random seeds (0, 25, 28, 30, 40, 50, 75, 100) to assess performance stability. For each seed, I measured Accuracy, Precision, Recall, and F1-score, and then computed the average values to capture overall model consistency. This analysis showed that ensemble-friendly models such as Random Forest and XGBoost maintained high stability across different splits, supporting their selection as reliable base learners for the stacking ensemble. Comparison Table Between Single Model and Hybrid Model with Random State seed value of [0, 25, 28, 30, 40, 50, 75, 100]

Average taken for Accuracy Of different Random State

Average Accuracy of Random State					
Model	**Hepatitis**	**Heart Disease**	**Diabetes**	**Liver Disease**	**Lung Cancer**
Logistic Regression	0.7864	0.8045	0.7927	0.8288	0.6979
Random Forest	0.8527	0.9103	0.8931	0.9047	0.8263
SVM	0.8310	0.8580	0.8569	0.8536	0.7899
KNN	0.8457	0.8334	0.8543	0.8564	0.8016
Stacking Hybrid Model	0.9028	0.9646	0.95485	0.9257	0.8997

Average Accuracy Of Random State Graph



5.5 Summary and Discussion

In this study, I developed a **stacking ensemble model** combining Random Forest, SVM (RBF), KNN, and XGBoost as base learners with Logistic Regression as the meta-learner, and used **Optuna** for hyperparameter tuning. The ensemble consistently outperformed individual classifiers across all disease datasets, achieving high accuracy, precision, recall, and F1-scores, while confusion matrices showed strong true positive and true negative predictions. Stability analysis across multiple random seeds confirmed the robustness of ensemble-friendly models like Random Forest and XGBoost. Overall, the results demonstrate that stacking ensembles effectively leverage complementary model strengths, providing a reliable and accurate multi-disease prediction framework with strong potential for real-world clinical application.

CHAPTER 6

CONCLUSION

In this study, I developed an optimized stacking ensemble model for multi-disease prediction. The model uses Random Forest, SVM, KNN, and XGBoost as base learners, with Logistic Regression as the meta-learner. This setup combines the strengths of different algorithms while reducing their weaknesses, resulting in more accurate predictions.

I evaluated the model on five disease datasets: Hepatitis, Heart Disease, Diabetes, Liver Disease, and Lung Cancer. The stacking ensemble consistently achieved high accuracy across all datasets. To ensure reliability, I tested the model using different random seed values, which confirmed that its performance is stable and generalizable.I also applied feature engineering, correlation analysis, and preprocessing to improve the model's effectiveness. Interaction features and risk scores helped capture important relationships between variables, while normalization and scaling ensured consistent input for the models. These steps not only improved accuracy but also made the predictions more interpretable.

Overall, the proposed stacking ensemble is robust, reliable, and generalizable, making it suitable for clinical decision support and early detection of multiple diseases. It demonstrates that combining multiple learners with careful feature preparation and preprocessing can significantly improve prediction performance.

CHAPTER 7

FUTURE ENCHANCEMENT

In the future, the project can be further improved and extended in several ways. Advanced feature selection techniques, such as SHAP (SHapley Additive Explanations) or Recursive Feature Elimination (RFE), can be employed to enhance model interpretability and better understand which features contribute most to disease prediction. Incorporating temporal or longitudinal data can help capture disease progression over time, enabling more accurate and personalized predictions. The stacking ensemble can be further optimized by dynamically adjusting ensemble weights, potentially improving overall predictive performance. Additionally, the model can be deployed as a real-time clinical decision-support tool with an intuitive interface for healthcare professionals, facilitating easier integration into routine medical practice. Testing the model on external datasets from different hospitals or regions will ensure robustness, generalizability, and broader applicability. Other potential enhancements include integrating multi-modal data such as medical images and lab reports, implementing automated hyperparameter tuning pipelines for continual improvement, and incorporating explainable AI techniques to increase trust and usability in clinical settings. These enhancements can make the system more accurate, reliable, and practical for real-world healthcare applications.

Furthermore, integrating real-time patient monitoring data, such as wearable device readings or continuous vital signs, can allow the model to provide timely alerts for early disease detection. Incorporating multi-center collaboration will enable the collection of diverse datasets, improving the model's robustness across different populations. Exploring hybrid models that combine machine learning with deep learning techniques, such as neural networks for imaging data, can further enhance predictive accuracy. Finally, implementing user feedback loops within the clinical interface can help the system learn from expert corrections, continuously improving its recommendations and ensuring practical usability in healthcare environments.

CHAPTER 8

BIBLIOGRAPHY

[1] J. Kim, H. Lee, and S. Park, “Ensemble Learning for Multi-Disease Diagnosis,” Journal of Medical Systems, vol. 43, no. 7, 2019.

[2] A. Rizvi, M. Khan, and S. Sharma, “Comparative Evaluation of Ensemble Approaches for Disease Prediction Using Multiple Datasets,” Health and Technology, vol. 11, no. 4, pp. 245–260, 2022.

[3] R. Sharma and S. Patel, “Hybrid Ensemble Framework Combining Random Forest, Gradient Boosting, and SVM for Multi-Disease Classification,” International Journal of Healthcare Informatics, vol. 9, no. 3, pp. 101–115, 2021.

[4] Y. Wang, Z. Li, and H. Chen, “Multi-Disease Prediction Using Machine Learning on Electronic Health Records,” IEEE Access, vol. 9, pp. 12345–12356, 2021.

[5] M. Khan, S. Rahman, and A. Ahmed, “Predicting Hypertension Using Machine Learning on Population Health Data,” Journal of Biomedical Informatics, vol. 118, pp. 103–115, 2022.

[6] A. Almasoud and P. Ward, “Predicting Chronic Kidney Disease Using Machine Learning with Feature Selection and Ensemble Optimization,” Journal of Medical Informatics, vol. 15, no. 2, pp. 78–91, 2021.

[7] A. Tiwari, R. Gupta, and P. Singh, “Ensemble Framework for Cardiovascular Disease Prediction,” International Journal of Computational Intelligence in Healthcare, vol. 8, no. 2, pp. 67–78, 2023.

[8] K. Patel and R. Mehta, “Hybrid Ensemble Model for Cardiovascular Disease Prediction Using Gradient Boosting, Random Forest, and Logistic Regression,” Journal of Clinical Data Science, vol. 12, no. 1, pp. 45–58, 2022.

[9] S. Rahman, A. Khan, and M. Hasan, “Voting-Based Ensemble Approach for Heart Disease Detection Using Clinical EHR Data,” Health Informatics Journal, vol. 30, no. 2, pp. 120–135, 2024.

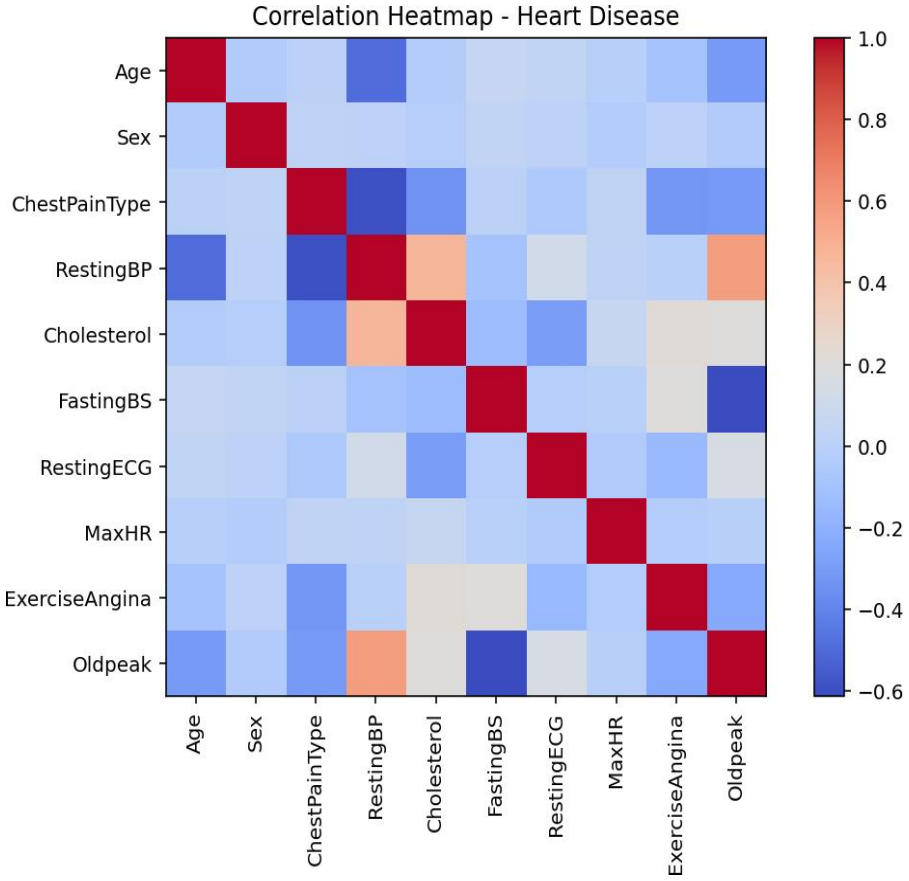
[10] V. Sharma, K. Patel, and R. Mehta, “Predictive Analytics for Multi-Disease Risk Assessment Using Machine Learning,” Health Informatics Journal, vol. 26, no. 3, pp. 202–218, 2020.

[11] X. Liu, L. Zhang, and Q. Chen, “Ensemble Risk Prediction Model for Multiple Chronic Diseases Using Clinical and Demographic EHR Data,” Journal of Biomedical Data Science, vol. 14, no. 4, pp. 89–102, 2021.

[12] M. Li, Y. Zhao, and Q. Chen, “Stacked Ensemble Learning for Chronic Disease Prediction,” Computers in Biology and Medicine, vol. 146, pp. 105–117, 2022.
[13] S. Kumar and P. Das, “Hybrid Stacked Ensemble Framework for Chronic Disease Risk Analysis Across Diabetes, Heart Disease, and Liver Disease Datasets,” International Journal of Healthcare Informatics, vol. 11, no. 2, pp. 56–70, 2023.
[14] Kaggle, “Kaggle Datasets,” [Online]. Available: https://www.kaggle.com/datasets.
[15] University of California, Irvine (UCI) Machine Learning Repository, “UCI Datasets,” [Online]. Available: https://archive.ics.uci.edu/ml/index.php.
[16] Spyder IDE, “Scientific Python Development Environment,” [Online]. Available: https://www.spyder-ide.org.
[17] Anaconda, “Anaconda Distribution for Python,” [Online]. Available: https://www.anaconda.com.
[18] Google Colab, “Collaborative Python Notebook Environment,” [Online]. Available: https://colab.research.google.com.
[19] Google Scholar, “Academic Research Database,” [Online]. Available: https://scholar.google.com.
[20] OpenAI, “ChatGPT: Large Language Model,” [Online]. Available: https://chat.openai.com.

APPENDIX

Heatmap Correlation of Dataset



Preprocessing dataset

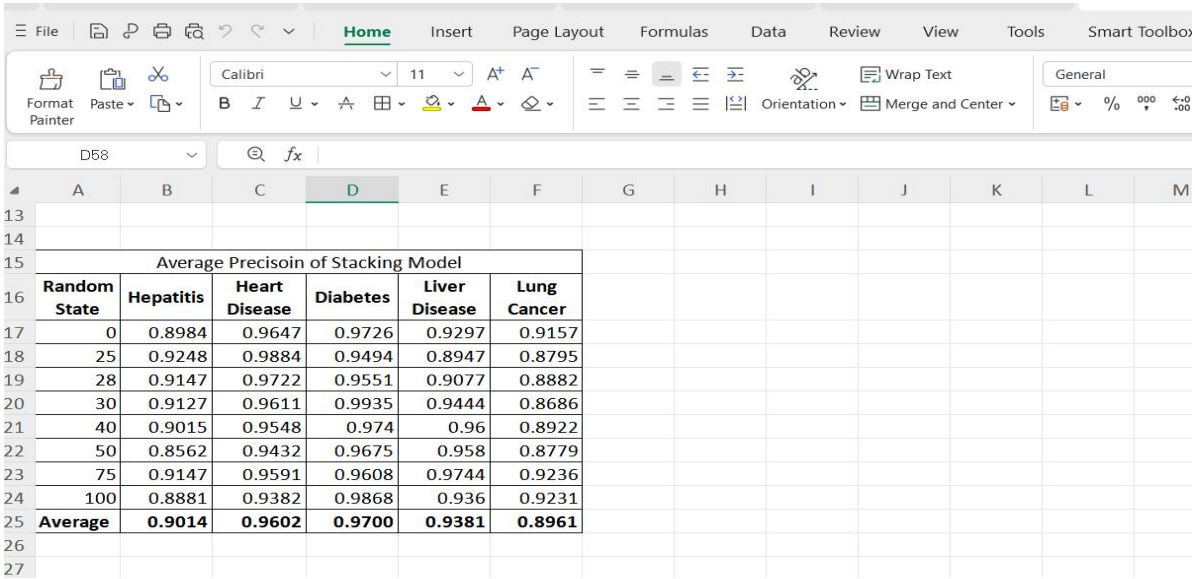
```
Preprocessing Hepatitis dataset...
Hepatitis preprocessed and saved to /content/drive/MyDrive/pg mini project/data/preprocess/hepatitis_preprocessed.csv
Preprocessing Heart Disease dataset...
Heart Disease preprocessed and saved to /content/drive/MyDrive/pg mini project/data/preprocess/heart_preprocessed.csv
Preprocessing Diabetes dataset...
Diabetes preprocessed and saved to /content/drive/MyDrive/pg mini project/data/preprocess/diabetes_preprocessed.csv
Preprocessing Liver Disease dataset...
Liver Disease preprocessed and saved to /content/drive/MyDrive/pg mini project/data/preprocess/liver_preprocessed.csv
Preprocessing Lung Cancer dataset...
Lung Cancer preprocessed and saved to /content/drive/MyDrive/pg mini project/data/preprocess/lung_cancer_preprocessed.csv
```

Output of the stacking model

```
Stacking Results with Confusion Matrix Breakdown:

               Accuracy  Precision  Recall  F1-Score     TP     TN    FP    FN
Hepatitis        0.9015     0.9038  0.9015    0.9014  114.0  124.0   8.0  18.0
Heart Disease    0.9717     0.9719  0.9717    0.9717  173.0  170.0   7.0   3.0
Diabetes         0.9710     0.9710  0.9710    0.9710  150.0  151.0   5.0   4.0
Liver Disease    0.9357     0.9372  0.9361    0.9357  114.0  119.0   4.0  12.0
Lung Cancer      0.8967     0.8969  0.8966    0.8966  150.0  145.0  19.0  15.0
```

Taken average of random state in excel



Average Precisoin of Stacking Model					
Random State	**Hepatitis**	**Heart Disease**	**Diabetes**	**Liver Disease**	**Lung Cancer**
0	0.8984	0.9647	0.9726	0.9297	0.9157
25	0.9248	0.9884	0.9494	0.8947	0.8795
28	0.9147	0.9722	0.9551	0.9077	0.8882
30	0.9127	0.9611	0.9935	0.9444	0.8686
40	0.9015	0.9548	0.974	0.96	0.8922
50	0.8562	0.9432	0.9675	0.958	0.8779
75	0.9147	0.9591	0.9608	0.9744	0.9236
100	0.8881	0.9382	0.9868	0.936	0.9231
Average	**0.9014**	**0.9602**	**0.9700**	**0.9381**	**0.8961**

SOURCE CODE

Preprocess Code

```
import pandas as pd

import numpy as np

import os

from sklearn.preprocessing import StandardScaler

from sklearn.impute import SimpleImputer

from sklearn.decomposition import PCA

from sklearn.feature_selection import RFE

from sklearn.ensemble import RandomForestClassifier

Paths to your raw synthetic datasets (update if needed)

datasets = { "hepatitis": "D:/sam/project/data/raw_hepatitis_dataset.csv",

"heart": "D:/sam/project/data/raw_heart_disease_dataset.csv",

"diabetes": "D:/sam/project/data/raw_diabetes_dataset.csv",

"liver": "D:/sam/project/data/raw_liver_disease_dataset.csv", "lung_cancer":
    "D:/sam/project/data/raw_formatted_lung_cancer_dataset.csv" }

============================================================

Step 1: Preprocessing function

def preprocess_dataset(file_path, save_name): df = pd.read_csv(file_path)

# -------- Feature Engineering Examples --------
    if "Age" in df.columns:
        # Age groups    df["Age_Group"] = pd.cut(df["Age"], bins=[0, 30, 50, 100],
```

```
                    labels=["Young", "Middle", "Old"])
    df["Age_Group"] = df["Age_Group"].astype(str)

if {"Cholesterol", "RestingBP"}.issubset(df.columns):
    # Risk score: Cholesterol / RestingBP
    df["Chol_BP_Ratio"] = df["Cholesterol"] / (df["RestingBP"] + 1)

if {"Weight", "Height"}.issubset(df.columns):
    # BMI calculation if weight/height exist
    df["BMI_Calc"] = df["Weight"] / (df["Height"] / 100) ** 2

# Encode categorical values
df = pd.get_dummies(df, drop_first=True)

# Target column (adjust if needed)
target_col = "disease_label"

# Separate features and target
X = df.drop(target_col, axis=1)
y = df[target_col]

# -------- Preprocessing --------
# Handle missing values
imputer = SimpleImputer(strategy="mean")
X = imputer.fit_transform(X)

# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# -------- Feature Selection --------
model = RandomForestClassifier(random_state=42)
rfe = RFE(model, n_features_to_select=int(X_scaled.shape[1] * 0.7))  # keep top 70%
X_selected = rfe.fit_transform(X_scaled, y)
```

```
    selected_features = np.array(df.drop(target_col, axis=1).columns)[rfe.support_]
    X_selected_df = pd.DataFrame(X_selected, columns=selected_features)

    # -------- Feature Extraction (PCA - optional) --------
    pca = PCA(n_components=min(10, X_selected.shape[1]))  # reduce to 10 or fewer
    X_pca = pca.fit_transform(X_selected)
    X_pca_df = pd.DataFrame(X_pca, columns=[f"PCA_{i+1}" for i in
    range(X_pca.shape[1])])

# -------- Final DataFrames --------
    df_selected = pd.concat([X_selected_df.reset_index(drop=True),
                             y.reset_index(drop=True)], axis=1)
    df_pca = pd.concat([X_pca_df.reset_index(drop=True),
                        y.reset_index(drop=True)], axis=1)
    # Save outputs
    out_file_selected = f"{save_name}_processed_selected.csv"
    out_file_pca = f"{save_name}_processed_pca.csv"

    df_selected.to_csv(out_file_selected, index=False)
    df_pca.to_csv(out_file_pca, index=False)

    print(f"Saved: {out_file_selected} and {out_file_pca}")
```

Run pipeline for all datasets

```
for name, file in datasets.items(): if os.path.exists(file): preprocess_dataset(file, name) else:
    print(f"File not found: {file}")
```

Stacking Model Code

```
# 1. Load Packages

import pandas as pd

from sklearn.model_selection import train_test_split, GridSearchCV

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

from sklearn.ensemble import StackingClassifier, RandomForestClassifier

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from sklearn.linear_model import LogisticRegression

from xgboost import XGBClassifier

from sklearn.metrics import confusion_matrix

# 2. Load Dataset Paths

datasets = {

    "Hepatitis": "U:/pcss23/project/data/preprocess/hepatitis_preprocessed.csv",

    "Heart Disease": "U:/pcss23/project/data/preprocess/heart_preprocessed.csv",

    "Diabetes": "U:/pcss23/project/data/preprocess/diabetes_preprocessed.csv",

    "Liver Disease": "U:/pcss23/project/data/preprocess/liver_preprocessed.csv",

    "Lung Cancer": "U:/pcss23/project/data/preprocess/lung_cancer_preprocessed.csv"

}

# 3. Define Hyperparameter Grids
```

```
param_grids = {
    "rf": {
        "n_estimators": [100, 200],
"max_depth": [None, 10, 20],
        "min_samples_split": [2, 5]
    },
    "svm": {
        "C": [0.1, 1, 10],
        "gamma": ["scale", "auto"],
        "kernel": ["rbf"]
    },
    "knn": {
        "n_neighbors": [3, 5, 7],
        "weights": ["uniform", "distance"]},
    "xgb": {
        "n_estimators": [100, 200],
        "max_depth": [3, 5, 7],
        "learning_rate": [0.01, 0.1, 0.2]
    }
}
```

```
# 4. Function: Tune Models

def tune_model(estimator, param_grid, X_train, y_train):

    grid = GridSearchCV(estimator, param_grid, cv=3, scoring="accuracy", n_jobs=-1, verbose=0)

    grid.fit(X_train, y_train)

    return grid.best_estimator

_

# 5. Function: Run Stacking with Tuning (with Confusion Matrix)

def run_stacking_with_tuning(dataset_path, target_column="disease_label"):

    print(f"\nLoading dataset: {dataset_path}")

    df = pd.read_csv(dataset_path)

    X = df.drop(columns=[target_column])

    y = df[target_column]

    # Train-test split

    X_train, X_test, y_train, y_test = train_test_split(

        X, y, test_size=0.2, random_state=30, stratify=y

    )

    # Tune each base learner

    tuned_rf = tune_model(RandomForestClassifier(random_state=42), param_grids["rf"], X_train, y_train)
```

```
    tuned_svm = tune_model(SVC(probability=True, random_state=42), param_grids["svm"],
X_train, y_train)
    tuned_knn = tune_model(KNeighborsClassifier(), param_grids["knn"], X_train, y_train)
    tuned_xgb = tune_model(
        XGBClassifier(use_label_encoder=False, eval_metric="logloss", random_state=42),
        param_grids["xgb"], X_train, y_train
    )
    # Base learners
    base_learners = [

        ("rf", tuned_rf),
        ("svm", tuned_svm),
        ("knn", tuned_knn),
        ("xgb", tuned_xgb)
    ]
    # Meta learner
    meta_learner = LogisticRegression(max_iter=1000, random_state=42)
    # Stacking model
    stacking_model = StackingClassifier(
        estimators=base_learners,
        final_estimator=meta_learner,
```

```
        cv=5,
        stack_method="predict_proba",
        n_jobs=-1
    )
    # Train and evaluate
    stacking_model.fit(X_train, y_train)
    y_pred = stacking_model.predict(X_test)
    # Compute metrics
    results = {
        "Accuracy": round(accuracy_score(y_test, y_pred), 4),
        "Precision": round(precision_score(y_test, y_pred), 4),


        "Recall": round(recall_score(y_test, y_pred), 4),
        "F1-Score": round(f1_score(y_test, y_pred), 4),
        "Confusion Matrix": confusion_matrix(y_test, y_pred)
    }
    print("\nConfusion Matrix:\n", results["Confusion Matrix"])
    return results
# 6. Run on All Datasets
final_results = {}
```

```
for name, path in datasets.items():
    print(f"\n=== Running Tuned Stacking on {name} Dataset ===")
    final_results[name] = run_stacking_with_tuning(path, target_column="disease_label")
# 7. Display Final Results
results_df = pd.DataFrame(final_results).T
print("\nStacking with Hyperparameter Tuning Results Across Datasets ===\n")
print(results_df.to_string())
```