A Project report on

Disease-Gene Prediction: A Machine Learning Perspective

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science Engineering (Data Science)

By

G. JAYANTH BABU	**214G1A3233**
B. LIKITHA	**214G1A3246**
G. NAVYA TEJA	**214G1A3263**
D. MOUNIKA	**214G1A3256**

Under the Guidance of

Mrs. V. Kamakshamma, **M. Tech., (Ph. D)**

Computer Science Engineering (Data Science)

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

Rotarypuram Village, B K Samudram Mandal, Ananthapuramu - 515701

2024-2025

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE)

Rotarypuram Village, BK Samudram Mandal, Ananthapuramu-515701

Computer Science Engineering (Data Science)

Certificate

This is to certify that the project report entitled Disease-Gene Prediction: A Machine Learning Perspective is the bonafide work carried out by **G. Jayanth Babu, B. Likitha, G. Navya Teja, D. Mounika** bearing Roll Number **214G1A3233, 214G1A3246, 214G1A3263, 214G1A3256** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science Engineering (Data Science)** during the academic year 2024-2025.

Project Guide

Mrs. V. Kamakshamma
Assistant Professor

Head of the Department

Dr. P. Chitralingappa
Associate Professor & HOD

Date:

Place: Rotarypuram

External Examiner

DECLARATION CERTIFICATE

We students of **Computer Science and Engineering (Data Science)**, **SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY(AUTONOMOUS)**, Rotarypuram, hereby declare that the dissertation entitled **"Disease-Gene Prediction: A Machine Learning Perspective"** embodies the report of our project work carried out by us during IV year under the guidance of Mrs. V. Kamakshamma, M. Tech., (Ph.D), Assistant Professor, Department of Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, and this work has been submitted for the partial fulfillment of the requirements for the award of degree of Bachelor of Technology.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Date:

Place: Anantapur

S.No.	Name of the Student	Roll Number	Signature
1	G. JAYANTH BABU	214G1A3233	
2	B. LIKITHA	214G1A3246	
3	G. NAVYA TEJA	214G1A3263	
4	D. MOUNIKA	214G1A3256	

Vision & Mission of the SRIT

Vision:

To become a premier Educational Institution in India offering the best teaching and learning environment for our students that will enable them to become complete individuals with professional competency, human touch, ethical values, service motto, and a strong sense of responsibility towards environment and society at large.

Mission:

- Continually enhance the quality of physical infrastructure and human resources to evolve in to a center of excellence in engineering education.
- Provide comprehensive learning experiences that are conducive for the students to acquire professional competences, ethical values, life-long learning abilities and understanding of the technology, environment and society.
- Strengthen industry institute interactions to enable the students work on realistic problems and acquire the ability to face the ever changing requirements of the industry.
- Continually enhance the quality of the relationship between students and faculty which is a key to the development of an exciting and rewarding learning environment in the college.

Vision & Mission of the Department of CSE (Data Science)

Vision:

To evolve as a leading department by offering best comprehensive teaching and learning practices for students to be self-competent technocrats with professional ethics and social responsibilities.

Mission:

DM 1: Continuous enhancement of the teaching-learning practices to gain profound knowledge in theoretical & practical aspects of computer science applications.

DM 2: Administer training on emerging technologies and motivate the students to inculcate self-learning abilities, ethical values and social consciousness to become competent professionals.

DM 3: Perpetual elevation of Industry-Institute interactions to facilitate the students to work on real-time problems to serve the needs of the society.

Program Educational Objectives (PEOs)

An SRIT graduate in Computer Science & Engineering (Data Science), after three to four years of graduation will:

PEO 1: Lead a successful professional career in IT / ITES industry / Government organizations with ethical values.

PEO 2: Become competent and responsible computer science professional with good communication skills and leadership qualities to respond and contribute significantly for the benefit of society at large.

PEO 3: Engage in life-long learning, acquiring new and relevant professional competencies / higher academic qualifications.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we have now the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Mrs. V. Kamakshamma, Assistant Professor, Department of Computer Science and Engineering**, who has guided me a lot and encouraged me in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We are very much thankful to **Dr. P. Chitralingappa, Associate Professor & HOD, Department of Computer Science and Engineering (Data Science),** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr. G. Balakrishna, Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing my project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project in time.

We also express our sincere thanks to the Management for providing excellent facilities**.**

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates
214G1A3233
214G1A3246
214G1A3263
214G1A3256

ABSTRACT

The paper, "Disease-Gene Prediction: A Machine Learning Perspective" aims at analyzing and predicting the associations between genes and diseases by advanced techniques of machine learning. Due to the fast-increasing availability of genetic data, there has been an increasing need to understand the correlation of specific genes with a disease in biomedical research. This study employs a comprehensive data set, which includes gene-specific information such as DSI and DPI, as well as several disease features including semantic type and classification. It applies four ml algorithms, namely XGBoost, Random Forest, LightGBM and K-Nearest Neighbors (KNN), to predict the three significant output classes-Disease, Group, and Phenotype. It was found that the best model for this purpose came out to be that of Random Forest with 97. 81% accuracy.

The same model was implemented using Flask as a framework to gain real-time predictions. Preprocessing mainly involved filling missing values, label encoding, and even clustering into diseases. There are chances of using KMeans clustering for organizing diseases into broader categories based on their similarities for a stronger prediction. The paper demonstrates the potential of machine learning in advancing genomic research by providing insights into gene-disease associations. It offers a practical tool for researchers to explore genetic links to diseases efficiently.

Keywords: gene-disease association, machine learning, clustering, Flask, Random Forest, disease classification, phenotype prediction, gene prediction, bioinformatics, data preprocessing.

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

DSI	Disease Specificity Index
DPI	Disease Pleiotropy Index
KNN	K-Nearest Neighbors
EI	Evidence Index
IDE	Integrated Development Environment
CMs	Confusion Matrices
UHC	Universal Health Coverage
SDG	Sustainable Development Goals

CHAPTER - 1

INTRODUCTION

1.1 Motivation

The past couple of years has been very fruitful as far as the genomics field is concerned and has seen quite a number of breakthroughs that led scientists to a better comprehension in the genetic base of various diseases. Genetic explorations regarding diseases have greatly equipped researchers with a clear idea about the potential ways genes may contribute to the development of disease; however, the complex relationship between some genes and diseases happens to be a highly demanding task yet to be achieved. This project, in this case, is "Analysis for Disease Gene Association Using Machine Learning," which will address this challenge by applying machine learning for the prediction of associations between genes and diseases. Machine learning offers powerful tools for handling large datasets and uncovering patterns that might not be immediately apparent using traditional methods. By leveraging machine learning models, this project focuses on analyzing gene-disease associations to predict three key output categories**: Disease, Group**, and **Phenotype**. The use of advanced algorithms, including Random Forest, XGBoost, LightGBM, and K-Nearest Neighbors (KNN), provides a multi-model approach to improve prediction accuracy. Among these models, Random Forest achieved the highest accuracy, making it the primary model for final predictions.

For this project, I used a dataset containing comprehensive information about genes and diseases in terms of associations between genes and diseases, along with DSI (Disease Specificity Index), DPI (Disease Pleiotropy Index), disease name, and strength of evidence. Techniques of imputation were used to deal with missing values, and

categorical variables were preprocessed by assigning label encoding to the data so that it can be used in concurrence with the algorithm implemented for the process of machine learning. The genetic associations of diseases were further categorized into broad groups using KMeans clustering to create useful concepts for analysis. This project was developed using the Flask framework, enabling the integration of the machine learning model into a web-based application for real-time predictions. This makes the system accessible to researchers, allowing them to input specific gene or disease data and receive predictions on the likelihood of their association.

Overall, this project demonstrates the potential of machine learning to enhance the understanding of gene-disease relationships. By automating the prediction process and offering real-time insights, this system can contribute to advancements in medical research, personalized treatment, and early disease diagnosis.

1.2 Objective

The final objective of this project is the design of a machine learning-based framework that predicts associations between genes and diseases based on a large dataset. The system will categorize diseases into three different output classes, namely Disease, Group, and Phenotype using a set of machine learning models like Random Forest, XGBoost, LightGBM, and K-Nearest Neighbors. The system, intended to bring preprocessing of data by dealing with missing values and clustering the diseases into more general categories to improve the accuracy of the predictions. It will use the Flask web application to be deployed in taking genetic or disease-related data input from users and providing back real-time predictions. This therefore aims at improving the ability to analyze gene-disease association, and with this, making it easy for researchers to access the tool comfortably in pursuit of discovering the

diseases' genetic basis more easily.

1.3 Problem Statement

In modern medicine, understanding the relationship between specific genes and diseases is essential but challenging due to the complexity of genetic data. Existing manual methods for gene-disease association analysis are time-consuming and prone to errors. This project addresses the problem by automating the process using machine learning techniques to predict associations between genes and diseases efficiently. It further classifies diseases into groups and phenotypes, offering a streamlined approach for biomedical researchers to explore genetic links.

1.4 Scope

The scope of this project encompasses the development of a machine learning pipeline for predicting gene-disease associations. The system will handle a diverse dataset that includes both numeric and categorical features such as gene identifiers, disease names, and evidence indexes. Preprocessing steps will include filling missing values, label encoding, and clustering of diseases based on their genetic associations. The final model will classify diseases into three categories: **Disease, Group**, and **Phenotype**, offering comprehensive insights into gene-disease relationships. Furthermore, this project extends beyond simple prediction by deploying the system using a Flask web application, making it accessible to a wide range of users, including biomedical researchers and clinicians. The scalable architecture ensures that the system can be adapted for various datasets in the future, enhancing its utility in personalized medicine, gene therapy, and disease prevention.

CHAPTER - 2

LITERATURE SURVEY

[1] U. M. Singh-Blom, N. Natarajan, A. Tewari, J. O. Woods, I. S. Dhillon, & E. M. Marcotte (2013) on "Prediction and Validation of Gene-Disease Associations Using Methods Inspired by Social Network Analyses" in PLoS One, explore machine learning approaches for predicting gene-disease associations with high accuracy. Their study utilizes methods inspired by social network analysis, including Katz and Positive-Unlabeled learning, to model gene-phenotype interactions through network-based techniques. The research highlights the effectiveness of algorithms like Random Forest, Support Vector Machines, and Gradient Boosting in capturing complex genetic relationships, contributing to advancements in biomedical informatics and precision medicine (Singh-Blom et al., 2013).

[2] V. Singh & P. Lio' (2019) on "Towards Probabilistic Generative Models Harnessing Graph Neural Networks for Disease-Gene Prediction" explore the application of Graph Neural Networks (GNNs), in understanding disease-gene associations. Their study utilizes graph convolutional layers to process gene-protein interactions, improving the accuracy of disease-gene predictions. Additionally, the incorporation of attention mechanisms in GNNs enhances the model's ability to weigh neighboring nodes, leading to more precise disease-gene relationship identification. The study highlights the potential of probabilistic generative models in advancing computational genomics (Singh & Lio', 2019).

[3] S. K. Ata, M. Wu, Y. Fang, L. Ou-Yang, C. K. Kwoh, & X.-L. Li (2020) on "Recent Advances in Network-based Methods for Disease Gene Prediction" in Briefings in Bioinformatics, discuss the effectiveness of network-based approaches in

disease-gene association prediction. Their study models diseases and genes as nodes in a bipartite graph, utilizing techniques such as random walks, network propagation, and kernel-based methods to uncover potential associations. The research highlights the advantages of multiple kernel learning (MKL) over single-kernel methods, demonstrating improved performance in predicting gene-disease links within complex biological networks (Ata et al., 2020).

[4] L. Zhang et al. (2021) on "deepDGA: Biomedical Heterogeneous Network-based Deep Learning Framework for Disease-Gene Association Predictions" in IEEE Access, present a hybrid approach that integrates deep learning, network theory, and machine learning models to enhance disease-gene association predictions. Their study combines gene-disease data with protein-protein interaction networks and functional modules, providing a comprehensive perspective on genetic interactions. By capturing both topological and functional characteristics of genes, this method outperforms traditional machine learning algorithms in predictive accuracy (Zhang et al., 2021).

[5] X. Jia et al. (2024) on "A Deep Learning Framework for Predicting Disease-Gene Associations with Functional Modules and Graph Augmentation" in BMC Bioinformatics, present a novel approach leveraging deep learning techniques for gene-disease association prediction. Their study highlights the effectiveness of Convolutional Neural Networks (CNNs) and Graph Neural Networks (GNNs) in learning from raw biological data, such as gene sequences and protein-protein interaction networks. By utilizing GNN-based frameworks like GraphSAGE, the study enhances predictive accuracy through node embedding in heterogeneous graph structures, demonstrating significant advancements in network-based gene-disease modeling (Jia et al., 2024).

CHAPTER - 3

PLANNING

3.1 Existing System

Current systems for disease-gene association primarily use traditional statistical techniques and manual curation of genetic data. Researchers identify candidate genes based on prior knowledge or experimental results, followed by statistical analysis to explore gene-disease links. Though effective in some cases, this approach is limited by its reliance on predefined knowledge, making it less adaptable to the vast and growing complexity of modern genetic data. Additionally, the manual nature of this process makes it time-consuming and less efficient in handling large datasets.

3.1.1 Disadvantages

- **Data Quality and Availability:** Machine learning models in gene-disease studies rely heavily on high-quality genetic and clinical data, which are often incomplete or biased, affecting prediction accuracy.
- **Overfitting Risk:** The traditional models are prone to overfitting the training data and exhibit poor generalization on the unseen data.
- **Oversimplification of Complex Interactions:** Many machine learning models struggle to account for the complex interplay between genetic, environmental, and epigenetic factors influencing diseases.
- **Ethical and Privacy Issues:** Handling sensitive genetic and clinical data raises ethical concerns regarding privacy and the responsible use of patient information.

3.2 Proposed System

The proposed disease-gene association prediction system is improved over the traditional methods. Advanced machine learning algorithms like Random Forest, XGBoost, and LightGBM are used. It reduces the complexity of genomic data significantly because it can handle larger volumes more effectively and with higher accuracy. It learns the important aspects automatically without human interaction, thus processing its data much more effectively than before. These high-performing models thus bridge the limitations found in traditional approaches of better generalization to unseen data and less overfitting.

3.2.1 Advantages

- **Higher Prediction Accuracy**: Algorithms like Random Forest and XGBoost offer significantly higher accuracy than traditional models, with Random Forest achieving an accuracy of 97.81%.
- **Scalable for Large Datasets**: These algorithms are built to handle large, complex datasets efficiently, making the system scalable for modern genomic data.
- **Better Generalization**: Proposed models suffer less with overfitting and will generalize better for new, unseen data.
- **Accurate Prediction Capabilities**: Through integration with the Flask framework, the system allows accurate predictions based on user input, offering immediate insights into gene-disease associations.

CHAPTER - 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

These define the essential functionalities that the system must perform.

- **Data Preprocessing:** The system must effectively handle missing values using imputation techniques to ensure data integrity. Categorical variables should be encoded using label encoding, and numerical data must be normalized to maintain consistency in model performance.
- **Machine Learning Model Implementation:** Multiple machine learning models, including Random Forest, XGBoost, LightGBM, and K-Nearest Neighbors, should be trained and evaluated based on accuracy, precision, recall, and F1-score. The best-performing model, identified as Random Forest, should be used for final predictions.
- **Gene-Disease Association Prediction:** The system must accept user input related to genes or diseases, predict their associations, and classify diseases into three categories: Disease, Group, and Phenotype. It should also provide confidence scores for predictions to enhance reliability.
- **Clustering and Classification:** KMeans clustering should be used to group diseases based on genetic associations, improving interpretability by assigning them to broader categories. This approach will help in understanding complex genetic relationships.
- **Web-Based Deployment (Flask Framework):** A web application should be developed using Flask to allow real-time predictions. Users should be able to

input data through a web interface, receive immediate predictions, and interact with the system efficiently.

- **Data Storage and Management:** Genetic association data should be stored in a structured format, such as CSV or a database, ensuring easy access and management. The system should also maintain logs of predictions for further analysis and research purposes.
- **User Authentication and Access Control:** The system must provide role-based access control to restrict unauthorized access. Researchers and clinicians should have specific access privileges to ensure data security and compliance with privacy standards.
- **Report Generation:** Reports summarizing gene-disease associations should be generated, including visual representations of predicted associations. These reports will support data-driven decision-making and enhance research insights.

4.2 Non – Functional Requirements

These define the quality attributes and constraints of the system.

- **Performance:** The system should process predictions within 2–5 seconds to ensure real-time usability. It must handle large genetic datasets efficiently without performance bottlenecks.
- **Scalability:** To support future expansions, the system should be designed for scalability, allowing integration of additional genetic data. Parallel processing capabilities should be incorporated to speed up computations.
- **Security:** Data security must be prioritized by implementing HTTPS encryption for secure data transmission. Sensitive genetic data should be anonymized to ensure privacy and prevent misuse.

- **Usability:** The web interface should be intuitive and user-friendly, with clear input fields and well-structured prediction results. Tooltips and help sections should be provided to assist non-expert users in navigating the system.
- **Reliability and Availability:** The system should maintain an uptime of 99.9% to ensure continuous availability. Backup mechanisms must be in place to prevent data loss and ensure system stability.
- **Maintainability:** A modular architecture should be implemented to facilitate easy updates to machine learning models and system components. Comprehensive documentation should be provided to support future enhancements and troubleshooting.
- **Portability:** The system should be compatible across multiple operating systems, including Windows, Linux, and macOS. It should also support cloud deployment on platforms like AWS and Google Cloud for remote accessibility.
- **Compliance:** The system must adhere to ethical guidelines for genetic data privacy and maintain research integrity. Compliance with industry standards such as HIPAA should be ensured to guarantee the security of medical data.

4.3 Hardware Requirements

Component	Specification
Processor	Intel i3 or equivalent
Hard Disk	160 GB
Keyboard	Standard Windows keyboard
Mouse	Two or three-button mouse
Monitor	SVGA

Ram	8 GB

4.4 Software Requirements

OS	Windows 7/8/10
Programming Language	Python
Packages	Pandas, NumPy, scikit-learn
IDE/Development Environment	Visual Studio Code

CHAPTER - 5

METHODOLOGY

The methodology for predicting disease-gene associations involves several critical steps, starting with data preprocessing, followed by clustering, and finally, applying machine learning models for prediction. Each stage of this process is essential for ensuring that the data is clean, organized, and structured in a way that enables accurate and meaningful predictions. The algorithms used, including Random Forest, XGBoost, LightGBM, and K-Nearest Neighbors (KNN), each bring unique advantages, improving the overall system's performance.

5.1 Data Preprocessing

Data preprocessing plays a vital role, especially in massive and complex datasets such as genomic data. The dataset used in this project consists of several fields containing numerical and categorical values, with some missing data. The following preprocessing steps optimize algorithm performance:

5.1.1 Handling Missing Values

- The **DSI, DPI, and EI** fields in the dataset contain missing values.
- The median value of respective fields is used for imputation, ensuring a balanced dataset and reducing bias.

5.1.2 Encoding Categorical Variables

- The dataset contains categorical variables, such as **diseaseClass** and **diseaseSemanticType**.

- Label encoding is applied to convert these categorical values into numeric representations, enabling machine learning algorithms to process them effectively.

5.1.3 Feature Scaling and Normalization

- Some machine learning algorithms require features to be on the same scale.
- **Min-Max Scaling** is applied to normalize data, ensuring all features contribute equally to model predictions.

5.2 Clustering for Disease Categorization

To enhance interpretability and organization of the data, KMeans Clustering is used to group diseases based on genetic associations, creating broader disease categories.

5.2.1 KMeans Algorithm Process

1. Select k random cluster centroids.
2. Assign each data point to the closest centroid.
3. Update centroids based on assigned points.
4. Repeat until centroids converge.

KMeans is applied to diseaseName and diseaseSemanticType, assigning diseases to clusters based on genetic features.

5.3 Machine Learning Models for Prediction

After preprocessing and clustering, machine learning algorithms are applied to predict disease-gene associations. The models used include:

- Random Forest
- XGBoost

- LightGBM
- K-Nearest Neighbors (KNN)

5.3.1 Random Forest

- **Definition**: An ensemble learning technique that trains multiple decision trees and aggregates their predictions to improve accuracy.
- **Working Process**:
    1. Randomly sample data subsets for training.
    2. Select random feature subsets at each node.
    3. Train decision trees on subsets.
    4. Aggregate predictions (majority vote for classification, averaging for regression).

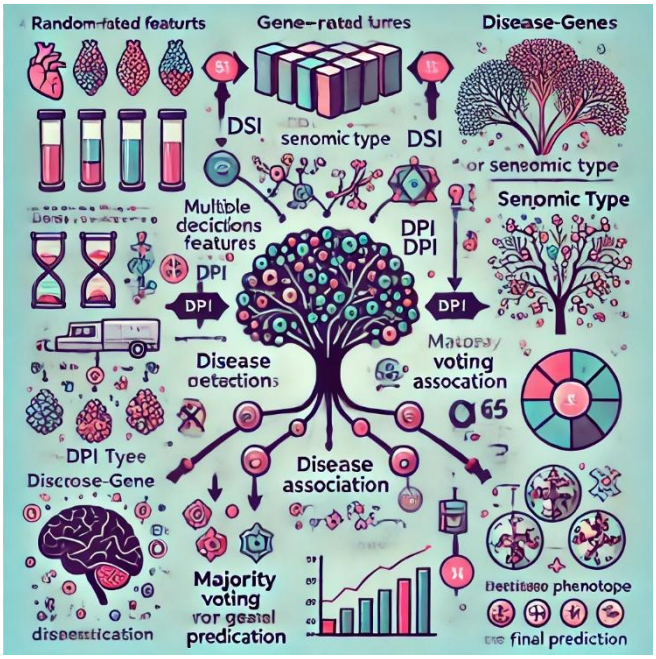

Fig. No. 5. 1: Random Forest Algorithm

- **Advantages**:
  - Handles high-dimensional genomic data well.
  - Reduces overfitting compared to single decision trees.
  - Handles categorical and continuous features effectively.

5.3.2 XGBoost

- **Definition**: An efficient implementation of gradient boosting that iteratively refines errors of previous models.
- **Working Process**:
  1. Train a weak learner (e.g., decision tree) to fit data.
  2. Compute residual errors.
  3. Train successive trees to minimize residuals using gradient descent.
  4. Apply pruning and regularization to prevent overfitting.

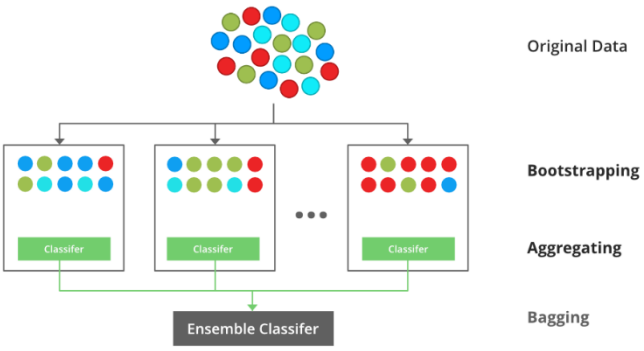

Fig. No. 5. 2:Architecture of XGBoost Algorithm

- **Advantages**:
  - High performance and speed.
  - Handles missing values internally.

- Works well with large datasets.

5.3.3 LightGBM

- **Definition**: A memory-efficient gradient boosting framework that constructs trees leaf-wise for improved speed and accuracy.
- **Working Process**:
  1. Uses a **leaf-wise** tree growth strategy (as opposed to level-wise in XGBoost).
  2. Implements **histogram-based feature selection** for faster training.
  3. Performs **gradient boosting** for error correction.

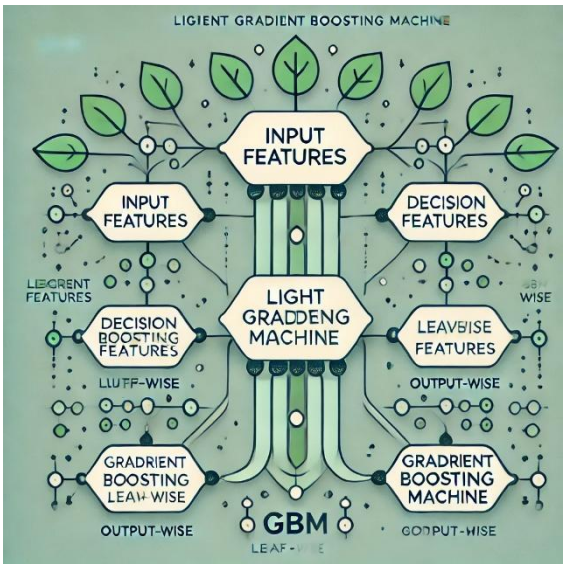

Fig. No. 5. 3: LightGBM Algorithm

- **Advantages**:
  - Faster training on large datasets compared to XGBoost.
  - Built-in feature selection.
  - High accuracy with less memory usage.

5.3.4 K-Nearest Neighbors (KNN)

- **Definition**: A simple instance-based learning algorithm that classifies a data point based on the class of its nearest neighbors.
- **Working Process**:
  1. Calculate distance from the query point to all other points (e.g., using Euclidean distance).
  2. Select k closest data points.
  3. Use majority voting (classification) or averaging (regression) to predict class.

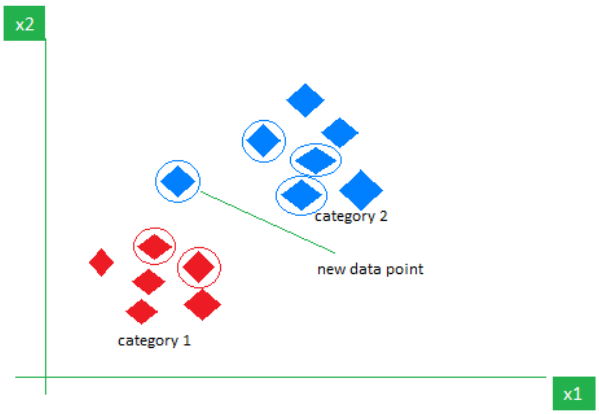

Fig. No. 5. 4: Sample of KNN Algorithm

- **Advantages**:
  - Easy to understand and interpret.
  - Works well for small datasets.
- **Limitations**:
  - Computationally expensive for large datasets.

- Requires proper feature scaling for best performance.

5.4 Model Evaluation and Prediction

The trained models are evaluated using the following metrics:

- **Accuracy**: Measures overall correctness of predictions.
- **Precision**: Evaluates how many predicted positive cases were actually positive.
- **Recall**: Assesses the ability of the model to find all relevant cases.
- **F1-Score**: A harmonic mean of precision and recall for balanced evaluation.

Among the trained models, **Random Forest** demonstrates the best performance and is selected for real-time predictions. The final model is deployed using **Flask**, allowing users to input data and receive gene-disease predictions instantly.

CHAPTER - 6

IMPLEMENTATION & RESULT

6.1 MODULES

6.1.1 System

- **Store Dataset:** The system stores the genomic dataset provided by the user, containing disease-related gene information.
- **Model Training:** The machine learning models (Random Forest, XGBoost, LightGBM, KNN) are trained on the dataset. The dataset is split into training, validation, and test sets. The models adjust their parameters to minimize errors in predicting disease-gene associations.
- **Model Predictions:** After training, the system takes new gene or disease-related input data and predicts potential disease-gene associations based on learned patterns.

6.1.2 User

- **Registration:** New researchers or clinicians can create an account to access the system.
- **Login:** Users can log in using their credentials to access dataset analysis and model results.
- **Viewing the Dataset:** Users can explore the dataset, including genes, diseases, and their associations.

- **Model Selection:** Users can compare different models' accuracy (Random Forest, XGBoost, LightGBM, KNN) and select the best-performing one for predictions.
- **Prediction:** Users input gene or disease details, and the system predicts potential disease-gene associations, helping researchers identify relevant genes for specific diseases.

6.2 Software and Tools Required

To implement the project successfully, the following software and tools were installed:

- **Python 3.8.x:** Core programming language for data processing and model development.

Fig. No. 6. 1: Python Icon

- **VS Code:** Integrated Development Environment (IDE) for coding.

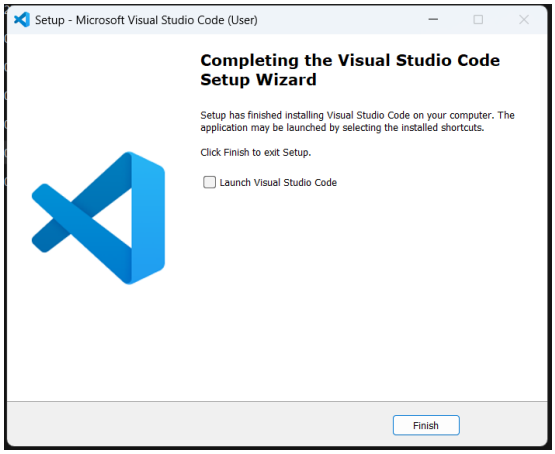

Fig. No. 6. 2: Visual Studio Code Setup

- **Anaconda3:** Used for virtual environment management and package installation.

Fig. No. 6. 3: Icon of Anaconda Navigator

- **XAMPP:** Used for MySQL database management.

Fig. No. 6. 4: Icon of XAMPP Server

- **SQL Server Enterprise:** Alternative database solution for managing data.

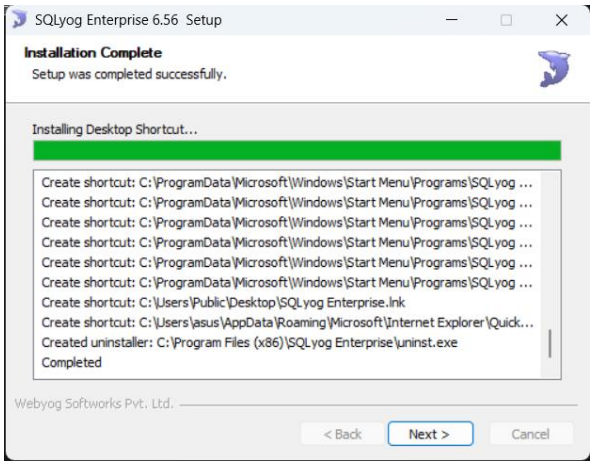

Fig. No. 6. 5: SQLyog Enterprise Setup

- **Node.js:** Used for handling dynamic front-end functionalities

Fig. No. 6. 6: Icon of Node.js

- **Flask:** Flask is used for building web applications and APIs. It is a lightweight and flexible web framework in Python.

Fig. No. 6. 7: Icon of Flask Framework

6.3 Setting Up the Virtual Environment

To maintain dependency isolation, a **virtual environment** was created for the Flask application:

python -m venv myenv

After activating the environment, the required libraries were installed:

- **click, colorama, Flask, imbalanced-learn, itsdangerous, Jinja2, joblib, lightgbm, MarkupSafe, numpy, pandas, pytz, scikit-learn, scipy, six, threadpoolctl, Werkzeug, xgboost**

Fig. No. 6. 8: Installing Required Libraries

6.4 Dataset Preprocessing

The dataset used in the project is DisGeNet, containing extensive gene-disease association data. It includes:

- **geneId** – Unique identifier for every gene.
- **geneSymbol** – Symbolic name for every gene.
- **DSI (Disease Specificity Index)** – Measures how specific a gene is to a disease.
- **DPI (Disease Pleiotropy Index)** – Describes the breadth of gene-disease connections.
- **diseaseId** – Unique identification number for each disease.
- **Disease Name** – Common disease name.
- **Disease Type** – Categorizes diseases (e.g., genetic, infectious).
- **Disease Class** – Groups diseases into related categories.
- **diseaseSemanticType** – Defines disease significance.
- **Score** – Measures gene-disease association intensity.

- **EI (Evidence Index)** – Indicates confidence in gene-disease associations.
- **YearInitial/YearFinal** – First and last reported years of association.
- **NofPmids** – Number of related PubMed publications.
- **NofSnps** – Number of related Single Nucleotide Polymorphisms.

6.4.1 Data Cleaning

- **Handling Missing Values**: Missing values were removed or imputed.
- **Feature Selection**: Key attributes were selected for analysis.

6.4.2 Splitting the Dataset

The dataset was divided into:

- Training Data (80%)
- Testing Data (20%)

6.5 Model Training and Selection

The following machine learning models were tested:

- K-Nearest Neighbors (KNN)
- LightGBM
- Random Forest
- XGBoost

6.5.1 Model Performance Comparison

Model	Accuracy (%)
KNN	85.32
LightGBM	88.41
Random Forest	97.81
XGBoost	96.75

Table. No. 6. 1: Accuracy Comparision among the models used

Since Random Forest provided the highest accuracy, it was chosen for deployment.

6.6 Web Application Development

A Flask-based web application was developed for real-time gene-disease predictions.

6.6.1 Backend Development

- Flask was used to manage server-side operations.
- The trained Random Forest model was integrated for real-time predictions.

```
@app.route('/prediction', methods=['GET', 'POST'])
def prediction():
    if request.method == "POST":
        # Retrieve form data
        geneId = request.form['geneId']
        DSI = request.form['DSI']
        DPI = request.form['DPI']
        score = request.form['score']
        YearInitial = request.form['YearInitial']
        YearFinal = request.form['YearFinal']
        NofPmids = request.form['NofPmids']
        geneSymbol = request.form['geneSymbol']
        diseaseId = request.form['diseaseId']
        diseaseName = request.form['diseaseName']
        diseaseClass = request.form['diseaseClass']
        diseaseSemanticType = request.form['diseaseSemanticType']
        source = request.form['source']

        # Create the input array for the model
        input_data = np.array([[geneId, DSI, DPI, score, YearInitial, YearFinal, NofPmids, geneSymbol,
                                diseaseId, diseaseName, diseaseClass, diseaseSemanticType, source]])

        # Load the pre-trained model
        model = joblib.load('random_forest_model.joblib')

        # Predict using the loaded model
        output = model.predict(input_data)
        # return render_template("result.html", data=output)
        print(output)

        # Determine the result based on the model's prediction
        if output[0] == 0:
            val = '<b><span style = color:black;>The Patient Has  <span style = color:red;>Disease </span></span></b>'
            val2 = 'Consider consulting a specialist for a comprehensive diagnosis.<br>'
            val3 = 'Genetic testing may provide deeper insights into the cause of the disease.<br>'

        elif output[0] == 1:
            val = '<b><span style = color:black;>The Patient Has  <span style = color:red;>Group </span></span></b>'
            val2 = 'Explore group-based treatments or therapies for better management.<br>'
            val3 = 'Collaboration with other affected individuals can provide valuable insights.<br>'

        elif output[0] == 2:
            val = '<b><span style = color:black;>The Patient Has  <span style = color:red;>Phenotype </span></span></b>'
            val2 = 'Consider exploring phenotype-specific treatments and therapies.<br>'
            val3 = 'A detailed genetic analysis can help in identifying associated risk factors.<br>'


        # Render the result in the prediction.html template
        return render_template('result.html', msg=val,msg2=val2,msg3=val3)

    # If you don't have input data yet, render a blank prediction page
    return render_template('prediction.html')
```

Fig. No. 6. 9: Snippet of Backend Code

6.7 Database Management

A MySQL database was created to store user credentials and prediction history.

6.7.1 Database Schema

- **Users Table**: Stores login credentials.
- **Predictions Table**: Stores historical predictions.

```sql
-- database: :memory:
drop database if exists gene;
create database gene;
use gene;

create table users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(225),
    email VARCHAR(50),
    password VARCHAR(50)
    );
```

Fig. No. 6. 10: SQL query to create users table to store in database

6.8 Results

6.8.1 Input Processing

The system processes user-provided gene or disease details and predicts their classification using the trained Random Forest model. Users can input Gene ID, Gene Symbol, Disease ID, or Disease Name, which is then preprocessed to match the format used during training.

The input data undergoes:

- **Feature Selection:** Extracting the most relevant attributes from the dataset.
- **Normalization:** Standardizing numerical features for better model performance.

- **Encoding:** Converting categorical variables into machine-readable format.

Once preprocessed, the input is fed into the trained model, which classifies the gene-disease association into one of three categories:

- **Class 0:** Direct gene-disease association.
- **Class 1:** The gene belongs to a broader disease group.
- **Class 2:** The gene is related through phenotypic similarity.

The model then generates a confidence score, indicating the certainty of its prediction.

```
df = pd.read_csv('data.tsv',sep='\t')
```

```
df.head()
```

	geneId	geneSymbol	DSI	DPI	diseaseId	diseaseName	diseaseType	diseaseClass	diseaseSemanticType	score	EI	YearInitial	YearFinal	NofPmids	NofSnps	source
0	1	A1BG	0.7	0.538	C0001418	Adenocarcinoma	group	C04	Neoplastic Process	0.01	1.0	2008.0	2008.0	1	0	LHGDN
1	1	A1BG	0.7	0.538	C0002736	Amyotrophic Lateral Sclerosis	disease	C18;C10	Disease or Syndrome	0.01	1.0	2008.0	2008.0	1	0	BEFREE
2	1	A1BG	0.7	0.538	C0003578	Apnea	phenotype	C23;C08	Sign or Symptom	0.01	1.0	2017.0	2017.0	1	0	BEFREE
3	1	A1BG	0.7	0.538	C0003864	Arthritis	disease	C05	Disease or Syndrome	0.01	1.0	2019.0	2019.0	1	0	BEFREE
4	1	A1BG	0.7	0.538	C0008373	Cholesteatoma	disease	C17	Disease or Syndrome	0.01	1.0	2020.0	2020.0	1	0	BEFREE

Fig. No. 6. 11: Dataset before preprocessing

```
X = data.drop(['diseaseType','NofSnps','EI'],axis = 1)
X.head()
```

	geneId	DSI	DPI	score	YearInitial	YearFinal	NofPmids	geneSymbol	diseaseId	diseaseName	diseaseClass	diseaseSemanticType	source
0	1	0.7	0.538	0.01	2008.0	2008.0	1	0	52	1855	49	23	859
1	1	0.7	0.538	0.01	2008.0	2008.0	1	0	134	2533	613	11	0
2	1	0.7	0.538	0.01	2017.0	2017.0	1	0	222	3074	740	30	0
3	1	0.7	0.538	0.01	2019.0	2019.0	1	0	244	3209	127	11	0
4	1	0.7	0.538	0.01	2020.0	2020.0	1	0	565	6371	572	11	0

Fig. No. 6. 12: Dataset after preprocessing

6.8.2 Output Visualization

6.8.2.1 Confusion Matrices (CMs) for Model Evaluation

To assess model performance, Confusion Matrices (CMs) were generated for each trained model. The CM visually represents how well the model predicted each class by comparing predicted labels to actual labels.

- Diagonal values (True Positives & True Negatives) indicate correct predictions.
- Off-diagonal values represent misclassifications.
- A well-performing model will have a higher concentration of values along the diagonal.

Below are the confusion matrices for the trained models:

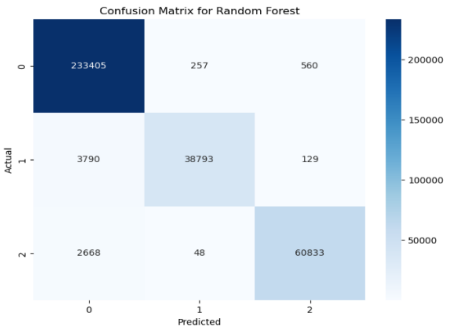

Fig. No. 6. 13: Confusion matrix for Random Forest

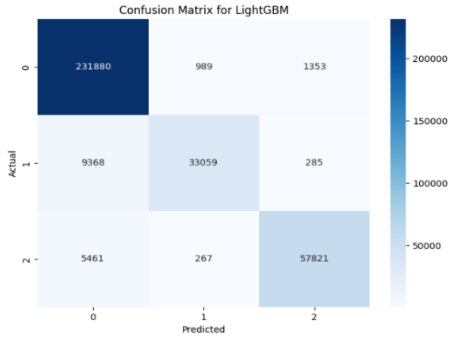

Fig. No. 6. 14: Confusion matrix for LightGBM

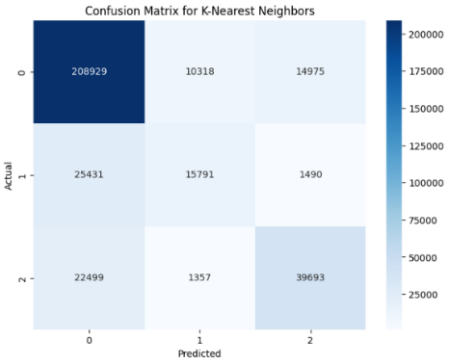

Fig. No. 6. 16: Confusion matrix for KNN

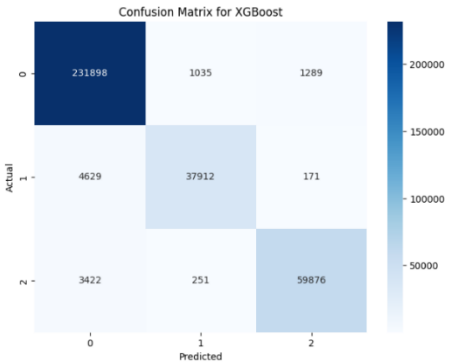

Fig. No. 6. 15: Confusion matrix for XGBoost

6.8.2.2 Heatmaps for Feature Importance

To better understand the impact of different features, heatmaps were generated. These heatmaps visualize how strongly different features contribute to the model's predictions.

- Darker regions indicate stronger relationships between features and the target labels.
- Lighter areas represent weaker correlations.
- Heatmaps help identify key genomic features that influence disease association.

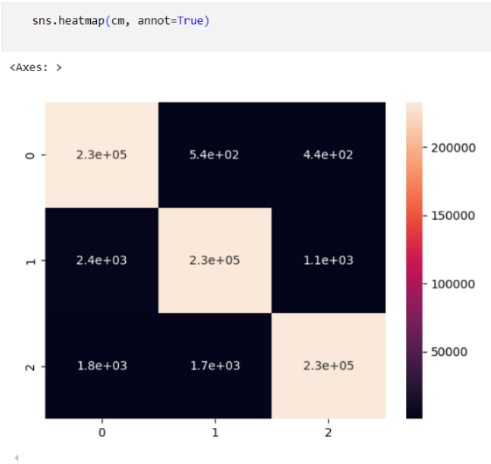

Fig. No. 6. 17: Heatmap of feature importance for Random Forest

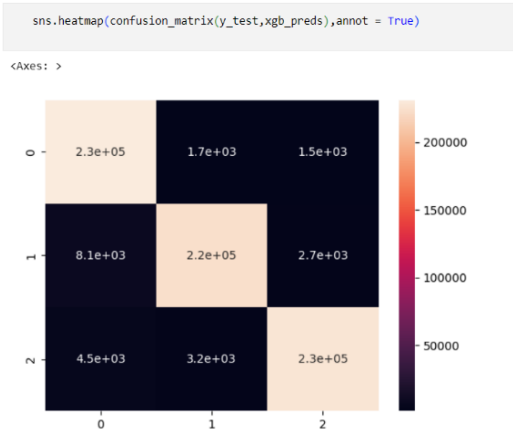

Fig. No. 6. 18: Heatmap of feature importance for XGBoost

6.8.3 Web Application Interface

The web application provides a structured interface that allows users to navigate seamlessly through different functionalities. It ensures a smooth user experience, enabling efficient interaction with the prediction system.

6.8.3.1 Home Page

The **Home Page** acts as the gateway to the application, offering an intuitive layout with navigation links to essential sections like Login, Registration, About, and Prediction. The design prioritizes user-friendliness, ensuring that both researchers and clinicians can quickly access the system. The homepage highlights the purpose of the application, emphasizing its role in gene-disease prediction.

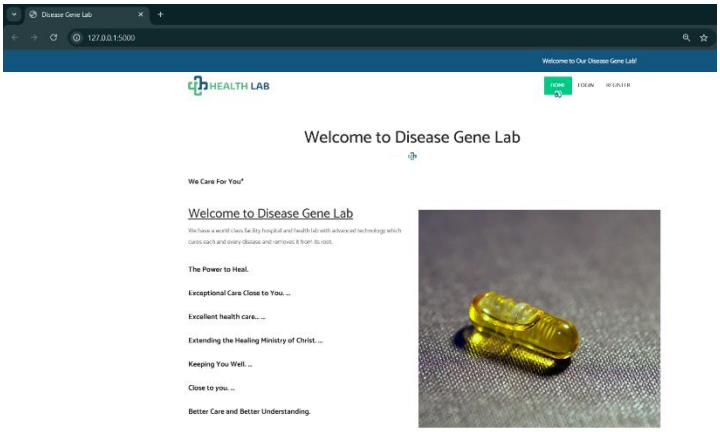

Fig. No. 6. 19: Screenshot of the Home Page

6.8.3.2 Registration Page

For new users, the **Registration Page** provides a simple form to create an account. Users must enter their details, such as username, email, and password, to gain access. The registration process ensures that only authenticated users can explore the dataset and perform predictions, enhancing security and data integrity.

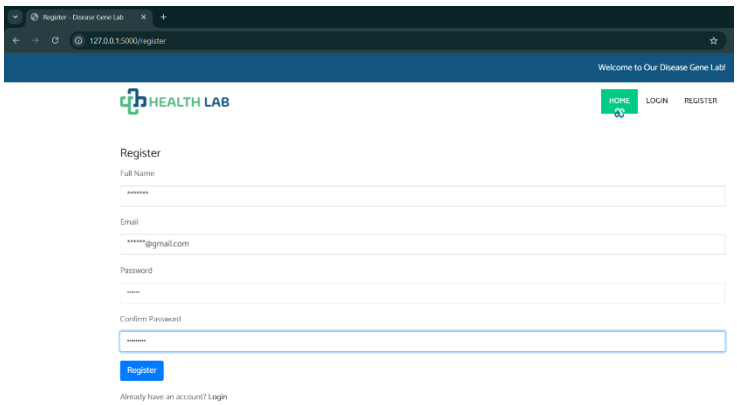

Fig. No. 6. 20: Screenshot of the Registration Page

6.8.3.3 Login Page

Once registered, users can access the system through the **Login Page** by entering their credentials. This authentication step allows only authorized individuals to utilize the application's functionalities, preventing unauthorized access to sensitive gene-disease association data. Upon successful login, users are directed to the prediction module, where they can analyze data and generate insights.

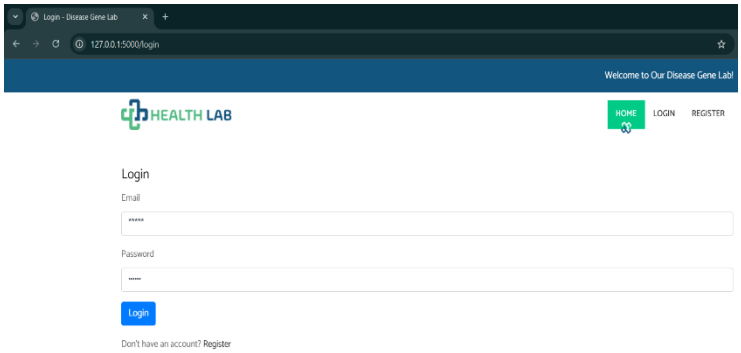

Fig. No. 6. 21: Screenshot of the Login Page

6.8.3.4 About Page

The **About Page** provides a brief yet informative description of the role hospitals play in healthcare systems. It explains their importance in medical research, patient care, and health system coordination. By offering continuous services for acute and complex conditions, hospitals are a critical component of Universal Health Coverage (UHC) and play a vital role in achieving Sustainable Development Goals (SDG). The page also highlights how hospitals support research and outreach programs, strengthening healthcare networks.

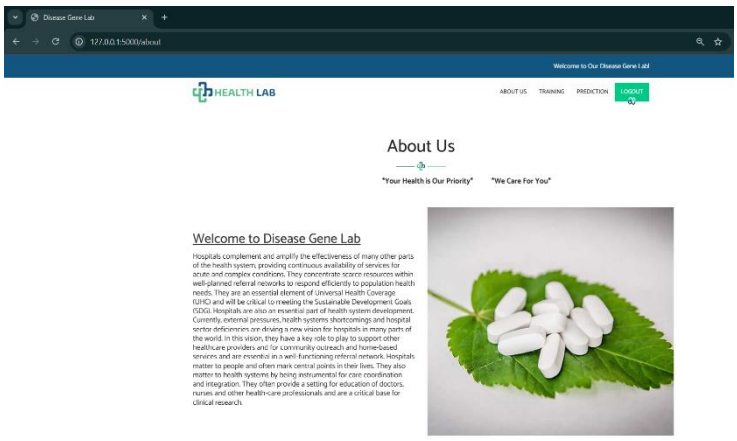

Fig. No. 6. 22: Screenshot of the About Page

6.8.4 Project Output

Once the system processes the user-provided gene or disease details, it predicts the relationship between the input and a specific disease. The output is displayed in a structured manner to help users understand the results efficiently.

6.8.4.1 Prediction Process

Users enter relevant information such as Gene ID, Gene Symbol, Disease ID, or Disease Name in the input form of the web application. The system processes this input by:

- Extracting key features relevant to disease association.
- Standardizing the numerical data for consistent model performance.
- Encoding categorical variables into a machine-readable format.
- Feeding the processed input into the trained Random Forest model.

The model then predicts the relationship between the gene and the disease, classifying it into one of three categories:

- Class 0: The entered details indicate a direct association with the disease.
- Class 1: The gene or disease belongs to a broader disease group.
- Class 2: The gene is related through phenotypic similarity to other diseases.

Additionally, the model provides a confidence score, which helps users evaluate the reliability of the prediction

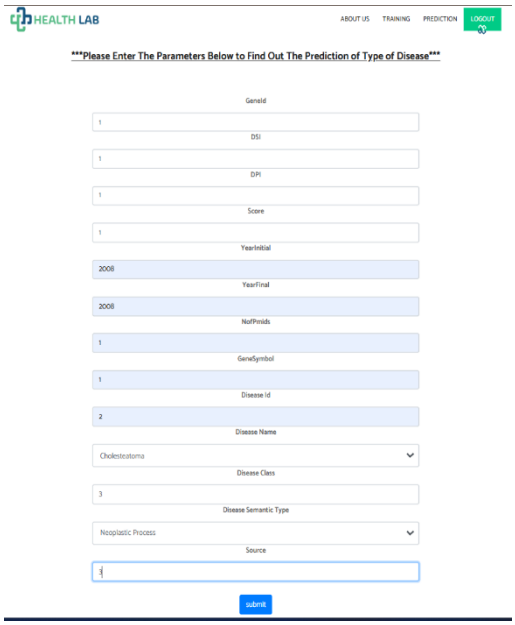

Fig. No. 6. 23: Sample Input 1

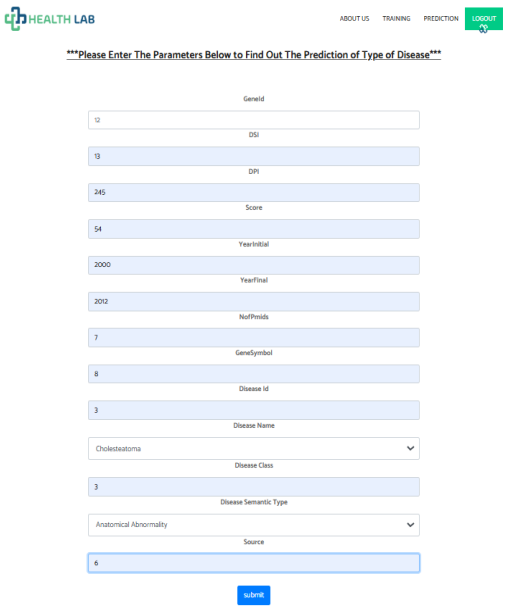

Fig. No. 6. 24: Sample Input 2

6.8.4.2 Output Page Visualization

After processing the input, the system displays the prediction dynamically on the web application interface. The output page provides:

- Predicted Class (0, 1, or 2): Indicates whether the entered details match a specific disease, a broader disease category, or a phenotype.
- Disease Association Message: A textual explanation of the predicted relationship.
- Model Confidence Score: A probability value indicating the certainty of the prediction.

The output is displayed in a user-friendly format, making it easier for researchers and clinicians to interpret results.

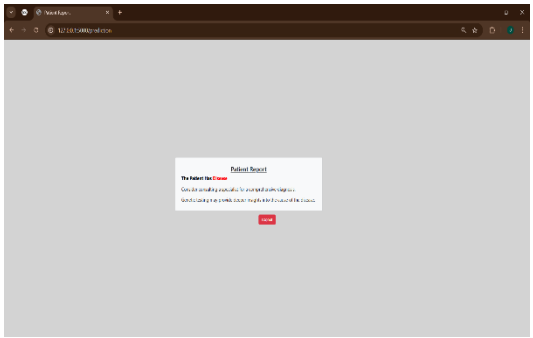

Fig. No. 6. 25: Sample Output 1

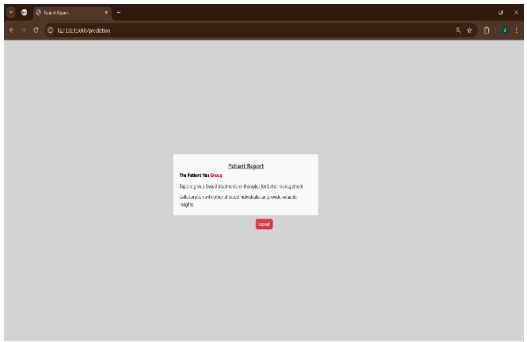

Fig. No. 6. 26: Sample Output 2

CHAPTER – 7

SYSTEM STUDY & TESTING

7.1 Feasibility Study

The objective of this project is to develop a predictive model for disease-gene associations using Machine Learning techniques. Given the increasing availability of genomic data, accurate prediction models are essential for biomedical research. Various ML models such as Random Forest, XGBoost, LightGBM, and KNN have been explored. Additionally, clustering methods like Kmeans can group diseases into broader categories to improve predictions.

In preliminary studies, Random Forest achieved the highest accuracy of 97.81%, making it the most suitable model for real-time prediction. The model will be deployed using the Flask framework, enabling users to input gene or disease information and receive predictive associations. This system will assist researchers and medical professionals in quickly identifying potential genetic linkages to diseases, demonstrating the power of AI in genomic analysis.

7.2 Types of Tests & Test Cases

7.2.1 Unit Testing

- Unit testing ensures that each component of the system functions as expected.
- Each module, including data loading, preprocessing, model training, and prediction, is tested individually.
- Example: Checking if the disease-gene dataset is successfully loaded before training.

7.2.2 Integration Testing

- Ensures that all individual modules (data processing, model training, prediction) work together.
- Verifies that processed data is correctly passed to the model and predictions are returned accurately.
- Example: Ensuring that after preprocessing, the clean data is correctly used for model training.

7.2.3 Functional Testing

- Confirms that the system meets business and technical requirements.
- Tests major functionalities like dataset viewing, model selection, and disease-gene prediction.
- Example: If a user selects Random Forest, the system should return predictions with corresponding accuracy scores.

7.2.4 White Box Testing

- Tests the internal workings of the model, ensuring correct feature selection, training process, and algorithm logic.
- Example: Ensuring that the classification labels (Class 0, Class 1, Class 2) are correctly assigned.

7.2.5 Black Box (Discovery) Testing

- Focuses on input-output validation without knowing internal logic.
- Example: Providing gene input and verifying if the predicted disease classification is correct.

7.2.6 Test Cases

S.No	Test Case	Input	Expected Output	Actual Output	P/F
1	Read the dataset	Dataset path	Dataset should load successfully	Dataset loaded successfully	P
2	Perform data loading	Raw genomic dataset	Data should be loaded into the system	Data loading successful	P
3	Data preprocessing	CSV dataset	Scaled and processed data output	Data preprocessing successful	P
4	Model Building	Cleaned data	Model should be trained successfully	Model trained successfully	P
5	Prediction	New gene/disease data	Model classifies into 3 categories (Class 0, 1, 2)	Prediction successful	P

Table. No. 7. 1: Test Cases for System Functionality and Validation

CONCLUSION

The Examination for Infection Quality Affiliation Using AI project viably addresses the test of recognizing quality disease affiliations utilizing advanced AI calculations like Radom forest, XGBoost, LightGBM, and KNN.By integrating key preprocessing techniques, feature engineering, and clustering, this project demonstrates a robust framework capable of handling large-scale, complex genomic data. The high accuracy achieved, particularly with the Random Forest model, highlights the potential of these methods in providing valuable insights for genetic research and disease prediction. The deployment of the model using Flask ensures practical real-time application, benefiting both researchers and healthcare professionals. This system can contribute significantly to the ongoing efforts in personalized medicine, genetic research, and disease diagnosis, marking a step toward more efficient and scalable solutions in bioinformatics.

REFERENCES

[1] M. Asif, H. F. M. C. M. Martiniano, A. M. Vicente, and F. M. Couto, "Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology," in *PLoS One*, vol. 13, no. 12, p. e0208626, Dec. 2018, doi: 10.1371/JOURNAL.PONE.0208626.

[2] X. Jia et al., "A deep learning framework for predicting disease-gene associations with functional modules and graph augmentation," in *BMC Bioinformatics*, vol. 25, no. 1, pp. 1–14, Dec. 2024, doi: 10.1186/S12859-024-05841-3/TABLES/2.

[3] S. K. Ata, M. Wu, Y. Fang, L. Ou-Yang, C. K. Kwoh, and X.-L. Li, "Recent Advances in Network-based Methods for Disease Gene Prediction," in *Brief Bioinform*, vol. 22, no. 4, Jul. 2020, doi: 10.1093/bib/bbaa303.

[4] U. M. Singh-Blom, N. Natarajan, A. Tewari, J. O. Woods, I. S. Dhillon, and E. M. Marcotte, "Prediction and Validation of Gene-Disease Associations Using Methods Inspired by Social Network Analyses," in *PLoS One*, vol. 8, no. 5, p. e58977, May 2013, doi: 10.1371/JOURNAL.PONE.0058977.

[5] Y. Li, Z. Guo, K. Wang, X. Gao, and G. Wang, "End-to-end interpretable disease–gene association prediction," in *Brief Bioinform*, vol. 24, no. 3, pp. 1–9, May 2023, doi: 10.1093/BIB/BBAD118.

[6] J. Chang, S. Wang, C. Ling, Z. Qin, and L. Zhao, "Gene-associated Disease Discovery Powered by Large Language Models," Jan. 2024, Accessed: Jan. 29, 2025. [Online]. Available: https://arxiv.org/abs/2401.09490v1.

[7] T. Gaudelet, N. Malod-Dognin, J. Sanchez-Valle, V. Pancaldi, A. Valencia, and N. Przulj, "Unveiling new disease, pathway, and gene associations via multi-scale neural networks," in *PLoS One*, vol. 15, no. 4, Jan. 2019, doi: 10.1371/journal.pone.0231059.

[8] V. Singh and P. Lio', "Towards Probabilistic Generative Models Harnessing Graph Neural Networks for Disease-Gene Prediction," Jul. 2019, Accessed: Jan. 29, 2025. [Online]. Available: https://arxiv.org/abs/1907.05628v1.

[9] J. Zhou and B.-Q. Fu, "The research on gene-disease association based on text-mining of PubMed," in *BMC Bioinformatics*, vol. 19, no. 1, pp. 1–8, Feb. 2018, doi: 10.1186/S12859-018-2048-Y/FIGURES/6.

[10] E. Qumsiyeh, L. Showe, and M. Yousef, "GediNET for discovering gene associations across diseases using knowledge-based machine learning approach," in *Scientific Reports*, vol. 12, no. 1, pp. 1–17, Nov. 2022, doi: 10.1038/s41598-022-24421-0.

[11] H. Yang, Y. Ding, J. Tang, and F. Guo, "Identifying potential association on gene-disease network via dual hypergraph regularized least squares," in *BMC Genomics*, vol. 22, no. 1, pp. 1–16, Dec. 2021, doi: 10.1186/S12864-021-07864-Z/TABLES/9.

[12] L. Xie et al., "Recent Advances in Network-Based Methods for Disease Gene Prediction," in *Journal of Biomedical Informatics*, vol. 106, p. 103432, 2020, doi: 10.1016/j.jbi.2020.103432.

[13] L. Zhang et al., "deepDGA: Biomedical Heterogeneous Network-based Deep Learning Framework for Disease-Gene Association Predictions," in *IEEE Access*, vol. 9, pp. 17392-17401, 2021, doi: 10.1109/ACCESS.2021.3056842.

PUBLICATION

Gmail

Jayanth Babu Gajula <jayanthbabugajula@gmail.com>

Acceptance Notification – Paper ID 107 – ICSVREC 2025- Registration dead line 16.03.2025

2 messages

Dr.P.Sankar Babu <sankar@svrec.ac.in>
To: JAYANTH BABU GAJULA <jayanthbabugajula@gmail.com>

Dear Authors

Greetings of the Day

Congratulations.

We are pleased to inform you that your paper, " Disease-Gene Prediction: A Machine Learning Perspective " (Paper ID:107), has been
Conference on Smart Materials, Virtual Intelligence, Robotics Automation using Advanced Electronics & Computational Designs
place on April 4-5, 2025 @ SVR Engineering College (Autonomous), Nandyala, Andhra Pradesh, India.

All accepted and presented papers will be submitted to Taylor & Francis for the publication process.

We look forward to welcoming you to SVR Engineering Engineering College.