# GENETIC DISORDER DETECTION USING MACHINE LEARNING – HEMOPHILIA B

Ms.N.Sumathi
Assistant Professor (SS)
Department of Computer Science and Engineering,
Dr. Mahalingam College of Engineering and Technology
Pollachi, Tamil Nadu, India
sumathinataraj@gmail.com

A Mohammed Rizwan
Department of Computer Science and Engineering,
Dr. Mahalingam College of Engineering and Technology
Pollachi, Tamil Nadu, India
amohammedrizwan@gmail.com

S. Divakar
Department of Computer Science and Engineering,
Dr. Mahalingam College of Engineering and Technology
Pollachi, Tamil Nadu, India
divakarselvam2003@gmail.com

P.K. Nithya Shree
Department of Computer Science and Engineering,
Dr. Mahalingam College of Engineering and Technology
Pollachi, Tamil Nadu, India
nithyashreepk03@gmail.com

**Abstract- A genetic disorder stems from abnormalities in DNA or alterations in chromosome number or structure. These conditions often result from mutations inherited from parents or arising spontaneously. Many well-known diseases are linked to these genetic mutations. Genetic testing plays a crucial role in helping individuals make informed choices regarding the prevention, treatment, or early identification of hereditary disorders. Research indicates a rising prevalence of genetic disorders alongside population growth, highlighting the need for continued study and intervention.**

**Hemophilia B is a hereditary bleeding disorder due to a deficiency in clotting aspect FIX, essential protein worried in blood clotting. This genetic situation commonly impacts adult males and may result in prolonged bleeding episodes even from minor injuries or spontaneous bleeding into muscle groups and joints. The severity of hemophilia B varies relying on the extent of issue F IX interest inside the blood. Causes of hemophilia B stem from mutations within the gene chargeable for generating aspect F IX, leading to its decreased or absent pastime. In integrating a genetic set of rules into Machine Learning venture, an initial population become created comprising various sets of hyperparameters for Support Vector Machine (SVM), Random Forest (RF), XG-Boost, K-Nearest Neighbor (KNN), and Naive Bayes classifiers. Every candidate solution changed into evaluated primarily based on its overall performance, represented with the aid of the accuracy rating attained on a validation dataset. Subsequently, the populace turned into scaled to prefer better-acting solutions, and a fitness feature tailored to every classifier was computed. Using genetic operations like crossover and mutation, new generations of answers were generated, refining the hyperparameter combos.**

**GA can optimize these algorithms by fine-tuning their parameters, helping them achieve better performance. Furthermore, GA can identify the most relevant features from a dataset, which can significantly improve model performance and efficiency. While GA is powerful, they require more computing power than traditional methods due to their iterative nature.**

## 1. INTRODUCTION

Genetic disorders result from abnormalities in an individual's DNA, often inherited from parents, leading to a spectrum of health challenges, from developmental hurdles to chronic ailments. These abnormalities can stem from mutations or environmental factors. Machine Learning (ML) algorithms have proven invaluable in comprehending and addressing genetic disorders by sifting through vast genomic data to detect.

Hemophilia B, colloquially known as Christmas disease, stands out as a rare genetic disorder characterized by a deficiency in clotting factor IX, a pivotal protein for blood coagulation. This insufficiency results in prolonged bleeding episodes, either spontaneously or post-injury. The condition arises from mutations in the gene responsible for producing factor IX, typically inherited on the X chromosome, hence its higher prevalence among males.

Data collection for Hemophilia B encompasses a wide array of information, including patient demographics, genetic profiles, bleeding patterns, treatment responses, and outcomes. This data is gleaned from diverse sources like medical records, genetic tests, patient registries, and ongoing research endeavours aimed at unravelling disease mechanisms, gauging treatment efficacy, and formulating personalized therapeutic strategies.

Genetic algorithms address these gaps by mimicking natural selection. They create a population of candidate solutions, like a group of individuals. he algorithm then selects the best solutions and mixes their traits to create new, improved solutions, mimicking reproduction. This cycle continues with each generation evolving closer to an optimal solution, following the concept of "survival of the fittest.

Furthermore, technological strides such as wearable devices and digital health platforms present opportunities for continuous monitoring and remote data collection, revolutionizing our comprehension and management of Hemophilia B. These innovations empower healthcare providers to monitor patients' conditions in real-time, furnishing invaluable insights into disease progression and treatment effectiveness, while streamlining the implementation of personalized care plans.

In essence, by harnessing machine learning algorithms and embracing technological advancements in data collection and analysis, we stand poised to enhance the diagnosis, treatment, and management of genetic disorders like Hemophilia B. Through the fusion of genomic data with clinical insights and the adoption of innovative technologies, we can deepen our understanding of these conditions and devise more targeted therapeutic interventions tailored to the unique needs of individual patients

## 2. RELATED WORKS

They added a brand-new approach to research the FIXa shape, as it should be predicting hemophilia B severity. The HemB-elegance framework efficiently forecasts mutation outcomes, assisting in clinical interpretation. Structural analysis identifies vital residues, guiding techniques. This method presents flexible tools for knowledge and managing hemophilia B and potentially different rare diseases. The examine hired supervised machine getting to know algorithms which includes decision tree, XGBoost, Random forest, and assist Vector device to predict the severity of Hemophilia B (HB) based totally on FIXa mutations. The models have been optimized using grid search and evaluated the use of validation strategies consisting of accuracy, Kappa Coefficient, Matthews Correlation Coefficient (MCC), and area underneath the ROC curve (AUC). The ensemble version, combining Random Forest and XGBoost, yielded the great consequences in phrases of accuracy and predictive performance**[1]**.

The brand-new modern imbalance-aware machine state-of-the-art techniques to predict deleterious genetic versions related to Mendelian and complicated diseases in non-coding areas. It employs a sampling approach wherein non-deleterious editions are randomly subsampled to lessen modern-day class imbalance, SMOTE are applied to increase the minority magnificence. Ensemble methods are then hired to combine predictions from a couple of models skilled on different subsets present day records, making sure insurance brand new to be had education information and diversity amongst base rookies. A hyper-ensemble technique is carried out, combining predictions from more than one random forest educated on unique balanced datasets. performance evaluation includes metrics like AUPRC and AUROC through cytobandconscious 10-fold pass-validation, ensuring unbiased trying out throughout chromosomal bands. The look at compares its hyper SMURF approach with state-of-the-art scoring techniques using numerous metrics, imparting a complete assessment contemporary predictive overall performance**[2]**.

The EAHAD Coagulation element variant Database assignment objectives to consolidate variant data associated with genes implicated in bleeding issues into a unified, web-reachable resource. It integrates curated structural, purposeful, evolutionary, and phenotypic information to resource in the classification of version pathogenicity. The assignment builds upon previous single gene variation databases, implementing new analysis gear, database architecture, and user interfaces. presently, it covers genes related to aspect VII (F7), issue VIII (F8), issue IX (F9), and Von Willebrand Factor (VWF), imparting complete records on genotype, phenotype (each laboratory and clinical), and the structural and practical impact of variants. This initiative enhances statistics high-quality and accessibility, facilitating more correct exams of disorder severity and pathogenicity within the haemostasis studies and scientific groups**[3]**.

The proposed method enhances multi-label multi-class genetic ailment prediction through GEDA for insights, characteristic engineering for excessive-importance characteristic selection, and ETRF for enriched function units. facts balancing guarantees equal elegance illustration, boosting model overall performance. Comparative analysis reveals sizeable overall performance improvements: and so forth's accuracy rises from 59% to 66% for label 1, whilst SVC's accuracy increases from 59% to 64%. furthermore, hamming loss decreases from 0.24 to 0.18, and the α-assessment rating increases from 86% to 91%. those findings underscore the effectiveness of the proposed technique in attaining higher accuracy and version performance. Comparative evaluation demonstrates sizable performance gains, with accuracy improvements and reduced hamming loss**[4]**.

In this studies article, a dataset comprising 22083 instances and 35 features was meticulously selected for genetic disorder prediction. using deep learning with artificial Neural Network (ANN), device studying techniques together with Support Vector Machine (SVM) and K-Nearest Neighbour (KNN) were hired for evaluation. thru rigorous preprocessing concerning information cleaning and function selection, the study performed a robust framework for accurate prediction of genetic disorders, leveraging the strengths of ANN, SVM, and KNN fashions. The Artificial Neural network (ANN) set of rules validated advanced overall performance all through the test. education accuracy stood at 85.7%, with a misclassification charge of 14.3% and an F1 rating of 92.2%. Validation accuracy reached 84.3%, with a misclassification rate of 15.7% and an F1 score of 91.3%. checking out accuracy became 84.9%, with a misclassification rate of 15.1% and an F1 rating of 92%. The validation Mean Squared Errors (MSE) became enormously low at zero.22, indicating high predictive accuracy**[5]**.

The proposed version utilizes gadget gaining knowledge of algorithms like random forests to expect genetic diseases across generations, enhancing accuracy and efficiency in comparison to traditional methods, thereby allowing proactive prevention of hereditary ailments. It predicts hereditary genetic sicknesses by using leveraging dataset analysis, enhancing prediction accuracy and performance metrics consisting of precision, and F1-measure to evaluate its effectiveness in sickness inheritance prediction. system getting to know classifiers are employed to predict hereditary developments, with enter facts present process pre-processing which includes dataset cleaning and label encoding. The venture encompasses loading facts, preprocessing, and classification using Random Forest, so It targets to expect hereditary genetic diseases and evaluates performance with accuracy, precision, F1-degree**[6]**.

This paper evaluated various preprocessing techniques on four category models, reading performance metrics across unique datasets. strategies covered SMOTE, under sampling, and a mixture of both with Tomek-links. decision tree version consistently outperformed others, exhibiting maximum balanced accuracy, consider, F1 rating, and AUC-ROC. notably, all fashions showed advanced performance on balanced statistics compared to the unique imbalanced dataset. Confusion matrices illustrated enhanced prediction of minority magnificence samples submit-preprocessing. KNN classifier done the best and F1 rating. Graphs depicting the evolution of function selection confirmed initial low balanced accuracy, which converged because the process advanced, indicating the effectiveness of the approach. The quality-acting populace changed into applied for very last predictions on check statistics**[7]**.

This paper gives the utility of Genetic Algorithms (GA) to robotically determine the trainable layers in switch CNNs. with the aid of encoding the variety of trainable layers as genes, the GA optimizes the transfer CNN structure across three datasets: cats_vs_dogs, horses or humans, and rock_paper_scissors. Consequences exhibit the efficacy of the GA in this task. moreover, insights from gradient evaluation provide in addition expertise of transfer AI models, even though decoding those models stays challenging. however, the method shows promise in advancing interpretability and explainability in AI models. moreover, DNA computing, leveraging DNA molecules for facts storage and molecular interactions for computation, gives parallelism blessings over digital computer systems, doubtlessly accelerating computation exponentially in certain cases**[8]**.

This research employs the quantile method for normalization and "normexp" for genetic prediction. However, drawbacks encompass restrained checking out space with only genes for class. The proposed technique, a hybrid technique, combines PCA, Regression, Random Forest algorithms to become aware of genetic versions related to disorder threat. PCA reduces dimensionality at the same time as preserving facts, Random Forest combines decision trees for category or regression, and decision trees break up nodes primarily based on parameters to create homogeneous sub-nodes, assisting in supervised machine learning knowledge of responsibilities. The P-GDA model is furnished the accuracy. The P-GDA model is furnished the accuracy as 97.34% and sensitivity as 96.45% for the GEO dataset. The accuracy of PGDA is higher as 3.9%and 6.17% than PCA and Random Forest algorithms respectively. The sensitivity is likewise outperformed as 2.2% and a couple of 8% than PCA and Random Forest algorithms respectively**[9]**.

This research ambitions to predict genetic issues using Machine Learning from scientific datasets, addressing the surge in hereditary disorders because of low genetic checking out cognizance amid population booms. For predicting genetic problems, K-Nearest Neighbour (KNN) and Cat Boost classifiers are utilized, at the same time as for subclass prediction, XGBoost, and Random Forest are employed. those algorithms are chosen for his or her effectiveness in handling high-dimensional data with elegance imbalances, ensuring most beneficial overall performance in type tasks. The accuracy of KNN is 60.59 in Classifier 1, the accuracy of KNN is 68.02 in Classifier 2**[10]**.

The data preprocessing completed converting missing values with column averages, enhancing dataset accuracy. Grid are seeking optimized SVM parameters, while the hybrid module combined genetic algorithms and SVM for function selection, boosting overall performance through parallel processing and genetic range upkeep. They carried out their Python set of rules on a quad-middle i7 processor with 8GB RAM and 1TB HDD. the usage of Scikit-research, Matplotlib, and NumPy, they evaluated their version on three datasets from UC Irvine. the usage of cell app, going via the ML set of guidelines at the cloud, carried out 75.9% accuracy at the Diabetes Dataset, decreasing features from 8 to 6. For the Liver Dataset, they attained 78.6% accuracy, decreasing capabilities from 10 to 8, with a slight loss as compared to using all functions**[11]**.

In this research paper, three ML fashions—Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN)—had been built for SLE prediction. A two-step SNP selection approach changed into employed to mitigate computational burden and overfitting. Random Forest and Support Vector Machine models were optimized for parameters including tree range and kernel functions, while ANN hyperparameters had been high-quality-tuned, ensuing in advanced predictive performance. Evaluation of supervised ML predictors (RF, SVM, ANN, and PRS) on a Chinese language SLE GWAS dataset discovered RF's superior performance (mean AUC = 0.84), surpassing different methods considerably. RF additionally exhibited better sensitivity (84%) and specificity (68%) at a most suitable reduce-off, with green computational time. Validation on

European populations showed RF's ability as an effective device for SLE class and early detection**[12]**.

The Genetic algorithm (GA) iteratively evolves severity prediction via health evaluation, selection, crossover, mutation, and survivor selection till convergence, optimizing solutions for complicated issues like genetic expression class in cancer RNA-Seq facts. A dataset with 802 samples and 21 genes across four clusters is categorized with 5 cancer sorts to deal with multiclass category, a divergent forest (DF) approach making use of Kulback Leible divergence is proposed, addressing limitations of Random forest's data benefit strategy. The DF classifier categorizes samples based totally on majority class votes after assessing records distribution differences the usage of KLD, enhancing classification accuracy in RNA-Seq data evaluation. The accuracy level generated using this is above 85%**[13]**.

A statistical analysis in comparison excessive and impartial f8 mutations, revealing sizable associations between unique parameters and HA prevalence ($p < 0.05$) features which includes conservation scores, Phosphorylation potential, MFE, GC ratio, nucleotide changes, codon utilization, and place in domain F5/8 kind A have been identified as predictive for HA-inflicting mutations. Decision tree, built on those parameters in predicting ailment occurrence, demonstrating the significance of both structural and sequence-based totally conservation stages in mutation analysis. A Decision tree model done 80% accuracy on F8 Test Set 1 (TP=290, FN=72) and 74% accuracy on F8 look at Set 2 (TP=324, FN=113). Comparative analysis with five prediction software tools, inclusive of PolyPhen-2 and SIFT-DNA, revealed similar overall performance in sickness prediction of hemophilia-causing mutations**[14]**.

Employing Python with Scikit-learn, Skfeature, and Hyperopt, we proven baseline studying methods, characteristic choice algorithms, and hyper-parameter optimization. effects, through ten-fold govalidation, show our approach outperforming SVM, and KNN across accuracy, precision, remember, and AUC. extensively, SVM advantages substantially from characteristic selection, notably with our genetic algorithm (GA) technique. This paper proposes a genetic set of rules wrapped SVM technique for Parkinson's disease detection, reaching advanced accuracy (0.95), precision (0.96), recall (0.98), and AUC (0.92) as compared to different techniques. Nine key functions are identified, improving SVM's performance. This method outperforms various feature choice techniques, showcasing its effectiveness in improving diagnostic outcomes for Parkinson's disease**[15]**.

# 3. METHODOLOGY

To initiate the analysis of mutations associated with Hemophilia B, pandas library was utilized for efficient data manipulation. This included loading, cleaning and pre-processing of the dataset. This dataset in particular pertains to Factor IX, a protein that is crucial for the blood clotting process and frequently mutated in cases of Hemophilia B. Visual representation is key to visualize how the protein structure gets altered due to the severity of Hemophilia B and for this the Matplotlib library was employed to plot the structural domains of Factor IX, along with the mutations illustrated on color dots on the protein structure.

This visualization helps in identifying the severity and distribution of mutations across various protein domains which ultimately provides for a clearer understanding of their potential impacts on protein function. The data was further prepared using Scikit-learn's utilities, which facilitated the splitting of the dataset into training and testing sets and the normalization of feature scales via the StandardScaler method. We applied multiple classification algorithms, including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), XGBoost, Random Forest, and Naive Bayes, all sourced from Scikit-learn. These models were trained to predict the severity of mutations from the features derived from the dataset.

To evaluate the efficacy of each model, we utilized Scikit-learn's metrics module to calculate accuracy, precision, recall, and F1-score. Additionally, the training and testing errors were examined to assess the generalization ability of the models across unseen data. This is imperative for understanding the performance and reliability of each model in real-world applications. Post initial model training and evaluation, the incorporation of genetic algorithm occurred to enhance model optimization. This algorithm, inspired by natural selection processes, is capable of performing complex searches for optimal feature subsets or hyperparameter configurations that could significantly enhance model performance.

In the context of our study, genetic algorithm could be utilized for selecting the most informative features or tuning model parameters to improve accuracy and efficiency. The mutations known to influence Hemophilia B was visualized on the structure of Factor IX. Each mutation's location and associated severity was highlighted to convey the potential impact on the protein's function. This is the overview of the methodology consisting of advanced data manipulation, robust machine learning techniques, and innovative optimization algorithms to study genetic mutations associated with Hemophilia B in Factor IX protein. Through this process of data visualization and comprehensive model evaluation, our approach enhances the understanding of the disease's genetic basis and also improves the predictive modeling of mutation impacts thereby eventually facilitating better clinical outcomes.

## 3.1 Data Collection module:

The dataset utilized in this research comprises a comprehensive array of bioinformatics and molecular biology parameters relevant to mutations in the Factor IX protein, which is significant in the context of hemophilia B. The data was meticulously compiled from several authoritative sources:

**AA_HGVS and AA_Legacy:** These columns specify the mutation names in HGVS (Human Genome Variation Society) nomenclature and their legacy names, respectively. Data was sourced from genetic mutation databases and literature.

**Protein_Change, aa1, and aa2:** Detail the specific amino acid changes due to mutations, with 'aa1' indicating the original amino acid and 'aa2' the new amino acid post-mutation. Information was extracted from genomic sequence analyses.

**AA_dist:** Represents the distance between the mutated amino acids, calculated using protein structural data.

**Psi and Phi:** These angles are part of the protein's secondary structure characterization, derived from crystallographic or NMR structure data.

**AreaSES and AreaSAS:** Surface area metrics computed from 3D protein models, indicating the solvent-exposed surface and solvent-accessible surface, respectively.

**RelSESA, kdHydrophobicity, and consurfDB:** Relate to the relative solvent-exposed surface area, the hydrophobicity of the amino acids, and conservation scores from the ConSurf database.

**Network Features (degree, betweenness, closeness, burts, pr, auth, kcore):** These are calculated from protein-protein interaction networks, indicating how mutations might affect molecular interactions.

**Predictive Scores (SIFT_score, Provean_score_2.5, Provean_score_0.05, Polyphen2 scores):** These are predictive metrics from computational tools assessing the impact of mutations on protein function and structure.

Within the analytical framework, this dataset forms the basis for training various machine learning models to predict the clinical severity of mutations in Factor IX : Feature Selection: Initial steps in the code involve selecting features that are most relevant to predicting mutation impacts. Techniques such as correlation analysis are employed to reduce dimensionality while retaining critical information.

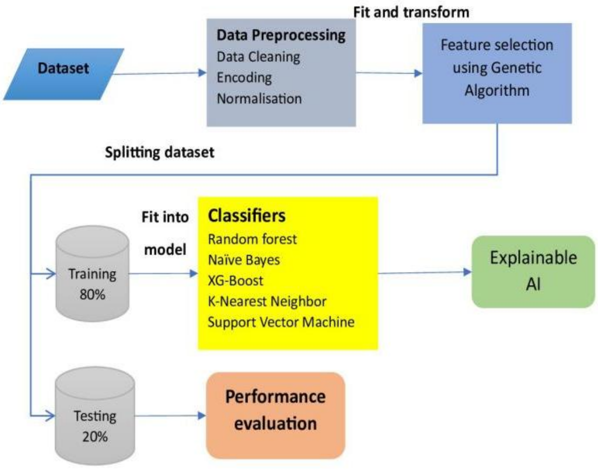


**Fig 3.1 : Methodology Diagram**

## 3.2. Data Preprocessing :

Data preprocessing is an important step before applying machine learning algorithms especially in the context of protein mutation severity classification which is the case with Haemophilia B. It ensures that the data is in a format suitable for the models to learn in the best way possible. This process involves the following steps :

Handling Missing Values :
Missing data points can be imputed using various techniques like mean/median imputation. In this case, it’s mean imputation such as

$$\mathrm{mean} = \frac{\sum_{i=1}^{n} x_i}{n}$$

n : Total number of data points in dataset
x : individual data points or values in the dataset

Encoding Categorical Variables:
Categorical features representing amino acids or other classifications need to be converted into numerical representations that machine learning models can understand. This involves label encoding.

$$\mathrm{dummy}(x_i) = \begin{cases} 1 & \text{if } x_i = \text{category} \\ 0 & \text{otherwise} \end{cases}$$

Feature Scaling:
Features often have different scales, which can bias the learning process. Techniques like normalization ( scaling features to a range like 0-1 ) or standardization are used to ensure all features contribute to the model’s learning. By executing these steps, we create a clean and standardized dataset which allows the machine learning algorithm to focus on identifying the patterns that differentiate between severe and non-severe mutations.

$$x_{std} = \frac{x - mean(x)}{std(x)}$$

x : Individual data point or observation from the dataset.

mean(x): Mean or average value of the entire dataset or distribution that the data point x belongs to.

std(x): This represents the standard deviation of the entire dataset or distribution.

## 3.3 Machine Learning Algorithm :

### 3.3.1 Random Forest Classifier

Random Forest is a learning method that condenses predictions together from multiple decision trees. Each tree is trained on a random subset of features and data points, increasing robustness and reducing variance compared to a single decision tree. While Random Forest doesn't have a single overarching formula, the core concept revolves around decision trees. Here's the formula for a decision tree split.

The final prediction of the Random Forest is made by majority vote from the individual trees' predictions.

**Algorithm : Random Forest**

```
Input : Test data
Ouput : Final Predicted class

1. Predict and store the outcome of each randomly created decision tree (T) based on given test data.
2. For each decision tree (T) in the Random Forest: a. For each data point x in the test data: i. Starting from the root node, traverse the tree: - At each decision node t: - Compute Feature_j(x) (the value of feature j for data point x) -
if Feature_j(x) ≤ threshold_t :
   Go to the left child node
else:
   Go to the right child node
3. Compute the total votes for each individual class.
4. Declare the majority class as the final outcome or classification.
```

### 3.3.2 Support Vector Machine

SVM is a classifier that maximizes the margin between the data points of different classes. Since the core optimization problem in SVM involves maximizing the margin, it can be formulated as below

Classifying new data points involves calculating the distance from the data point to the hyperplane using the equation:

If the decision function is positive, the data point is classified as class +1; otherwise, it's classified as class -1.

**Algorithm : Support Vector Machine**

**Input:** Training examples (X, y), kernel function K, regularization parameter λ
**Output:** Trained SVM model (w, b)

```
1. Initialize weight vector w = 0, bias term b = 0
2. For each training example ($x_i$, $y_i$) in (X, y):
3. Compute the predicted output:
   $y_{hat} = w^T * k(x_i, X) + b$
4. Calculate the hinge loss: loss = max(0, 1 - $y_i$ * $y_{hat}$)
5. Update the weight vector and bias term:
6. w = w + λ * $y_i$ * K($x_i$, X) if loss > 0
7. b = b + λ * $y_i$ if loss > 0
8. Return the trained SVM model (w, b)
```

### 3.3.3 XGBoost Classifier

XGBoost is a gradient boosting framework that builds an ensemble of decision trees sequentially. Each tree aims to correct the errors of the previous tree, leading to a more accurate model. XGBoost builds upon the concept of gradient boosting by minimizing an objective function that combines training loss and a regularization term to prevent overfitting.

XGBoost utilizes efficient algorithms to calculate gradients and update the model iteratively.

Relevant genetic features are extracted from patient genomes. Then, XGBoost is trained on this data to learn patterns indicative of genetic disorders. By analyzing these patterns, the model can accurately classify individuals with genetic disorders, aiding in diagnosis and personalized treatment strategies.

**Algorithm : XGBoost**

**ssInput:** Training data (X, y), number of trees M, learning rate α
**Output:** Ensemble of M decision trees

```
1. Initialize the predictions: $y_{pred} = 0$
2. For m = 1 to M:
3. Calculate the residuals: r = y - $y_{pred}$
4. Fit a decision tree fm to the residuals r, using X as features
5. Determine the weight for the current tree :
   $w_m = α$
6. Update the predictions:
   $y_{pred} = y_{pred} + w_m * f_m(X)$
7. Objective Function Minimization
   Objective(t) = Loss(t) + γ * T(t)
   Loss(t) = Training Loss(y, $y_{pred}$)
   T(t) = Complexity Term($f_m$)
8. Minimize Objective(t) to update the model parameters
9. Return the ensemble of M decision trees.
```

### 3.3.4 K-Nearest Neighbors (KNN)

KNN is a non-parametric lazy learning algorithm that classifies data points based on the labels of their k nearest neighbors in the feature space. KNN doesn't have a specific formula in the traditional sense. The classification process involves calculating the distance between a new data point (x) and each data point in the training set using a distance metric like Euclidean distance.

The k nearest neighbors of the new data point are identified based on the calculated distances. The most frequent class label among these k neighbors is assigned as the predicted class for the new data point.

Each genetic variation is represented as a point in a multidimensional space, where features correspond to genetic attributes. KNN assigns a class to a new variation by examining the classes of its nearest neighbors. This approach leverages the assumption that similar genetic variations tend to exhibit similar phenotypic traits or disorders. By analyzing the characteristics of neighboring variations, KNN aids in predicting the likelihood of a particular genetic disorder.

**Algorithm : K-Nearest Neighbors**

```
Input: Training data ($X_{train}$, $y_{train}$), new instance $x_{new}$, number of neighbors k
Output: Predicted class label for $x_{new}$

1.Initialize a list of distances D = []
2.For each training instance $x_i$ in $X_{train}$:
Calculate the Euclidean distance d($x_i$, $x_{new}$) between $x_i$ and $x_{new}$ using the formula:
        $d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$
  y : The i-th data point in the training set.
  $x_i$ and $y_i$: The i-th features of x and y, respectively.
Append d($x_i$, $x_{new}$) to the list D
3. Sort the list D in ascending order
4. Initialize a dictionary to store class counts: class_counts = {}
5. For each of the k nearest neighbors:
6. Return the class label with the highest count in class_counts.
```

### 3.3.5 Naïve Bayes Classifier

Naive Bayes is a probabilistic classifier based on Bayes' theorem. It assumes independence between features and calculates the posterior probability of a data point belonging to a particular class based on the individual feature probabilities. Bayes' theorem forms the foundation of Naive Bayes

$$\text{P(Class | Features)} = \frac{\text{P(Features | Class) * P(Class)}}{\text{P(Features)}}$$

**P(Class | Features):** The posterior probability of a data point belonging to a specific class given the observed features.

**P(Features | Class):** The likelihood of observing the features given the class.

**P(Class):** The prior probability of the class.

**P(Features):** The total probability of observing the features (marginal probability).

Naive Bayes calculates the likelihood of each feature value for each class independently and then multiplies them together using the product rule.

**Algorithm : Naive Bayes**

```
Input:
instance: The instance to classify (e.g., a list or vector of features)
class_probabilities: A dictionary containing the prior probabilities of each class
feature_likelihoods: A dictionary containing the likelihoods of features given each classss
Output:
    predicted_class: The predicted class for the given instance

Function NaiveBayesClassify (instance, class_probabilities, feature_likelihoods):
scores = {}
  for class in class_probabilities:
        scores[class] = log (class_probabilities[class])
      for feature in instance:
        if feature in feature_likelihoods[class]:
  scores[class] += log(feature_likelihoods[class][feature])
  predicted_class = max(scores, key=scores.get)
  return predicted_class
```

## 3.4. Optimization :

### 3.4.1. Model Training with Internal Hyperparameter Tuning

During the training phase of each machine learning algorithm, optimization occurs internally. This process involves the algorithm adjusting its hyperparameters, such as learning rates or tree depths, to minimize a predefined loss function, such as classification error. The goal is to find the optimal configuration that best fits the training data and minimizes errors in predicting mutation severity.

### 3.4.2. Hyperparameter Tuning with Genetic Algorithm (External)

This optimization process operates externally to the machine learning algorithms and relies on the utilization of a Genetic Algorithm (GA). The GA functions as an independent search mechanism, managing a population of potential configurations (hyperparameter sets) for each machine learning algorithm. These configurations undergo evaluation based on their corresponding model performances on a validation set. Through iterations, the GA applies selection, crossover, and mutation operations to refine configurations, aiming for enhanced performance. This iterative refinement persists until a predetermined stopping criterion, such as reaching the maximum iteration limit, is fulfilled. Subsequently, the most optimal configuration identified by the GA is employed to train a final model for subsequent evaluation.

### 3.4.3. Genetic Algorithm :

Incorporating a mutation genetic algorithm can enhance model performance by introducing diversity in the population of potential solutions. The algorithm iteratively evolves a set of candidate solutions, mimicking biological evolution, to search for an optimal or near-optimal solution. Mutation, a crucial component of genetic algorithms, introduces randomness by altering a small portion of the solutions.

For instance, random changes in the hyperparameters of your classifiers or perturb the features used for training. This process encourages exploration of different regions in the solution space, potentially discovering better-performing models or configurations that might not be reached through traditional optimization methods alone.

Mutation in genetic algorithms involves probabilistically selecting individuals from the population and applying random modifications to their genetic representation. These modifications can range from small perturbations to significant alterations, depending on the specific mutation operator and the problem domain. By diversifying the population through mutation, genetic algorithms can escape local optima and converge towards more globally optimal solutions.

```
Input: Data relevant to the problem
Output: Best individual with optimized fitness score
1. Initialization:
   - Define individual structure with genotype and fitness
   - Initialize population:
   - Create empty population list
   - Add individual to population list
2. Iterative Loop (Generations):
   - For each generation:
   - Evaluate fitness:
   - For each individual in population:
   - Calculate fitness score using input data and genotype
     - Select next generation:
     - Choose individuals from new population for next generation
3. Termination:
   - After set number of generations or stopping criteria met:
   - Find individual with best fitness score
   - Return best individual
```

**Genetic Algorithm Formulas :**

**Selection Probability**

$$\text{Probablity} = \frac{\text{Fitness}}{\text{Total Fitness}}$$

Explanation: To select individuals for reproduction, calculate the probability of each individual based on its fitness relative to the total fitness of the population.

Selection Probability plays a crucial role in genetic algorithms, determining the chances of each individual in the population being chosen for reproduction based on its fitness relative to the total fitness of the population. This entails computing the probability of selection for each individual, which is directly linked to its fitness compared to the overall fitness of all individuals. Higher-fitness individuals are assigned greater probabilities of selection, reflecting their potential to contribute valuable genetic material to the succeeding generation. By utilizing selection probabilities, genetic algorithms prioritize individuals with superior traits, thereby steering the evolutionary process towards solutions that better meet the objectives of the problem at hand.

**Crossover**

$$\text{OffSpring} = \frac{(\text{Parent1} + \text{Parent2})}{2}$$

Explanation: Combine genetic material from two parents to create offspring using techniques like single-point crossover or multi-point crossover.

Crossover is a fundamental operation in genetic algorithms where genetic material from two parent individuals is exchanged to produce offspring. It involves selecting a random crossover point along the chromosomes of the parents and swapping the genetic information beyond that point. This process results in the creation of new individuals, or offspring, with a combination of traits inherited from both parents. Common techniques include single-point crossover, where a single crossover point is chosen, and multi-point crossover, where multiple crossover points are selected. Crossover promotes genetic diversity within the population and facilitates the exploration of the solution space, ultimately enhancing the evolutionary process in genetic algorithms.

**Mutation Probability**

$$\text{Mutation Probability} = \frac{1}{(\text{Length of Chromosom}}$$

Explanation: Determine the probability of mutation for each gene in a chromosome, typically inversely proportional to the length of the chromosome.

Mutation probability is a crucial concept in genetic algorithms, determining the likelihood of genetic mutation occurring at individual gene positions within a chromosome. Typically, this probability is inversely proportional to the length of the chromosome, implying that shorter chromosomes have a higher likelihood of mutation compared to longer ones. It plays a vital role in introducing variability and diversity into the population, thus enabling exploration of the solution space. By adjusting mutation probabilities, researchers can tailor the balance between exploration and exploitation, enhancing the algorithm's effectiveness in finding optimal solutions.

**Mutation**

Mutated Gene = Gene + Random(Number between -Δ and +Δ)

Explanation: Introduce small random changes to genes in the chromosome to maintain diversity and explore new solutions.

Mutation is an essential process in genetic algorithms, vital for maintaining diversity and facilitating the exploration of new solutions within the population. It involves the introduction of random changes to genes in the chromosome. Specifically, each gene undergoes mutation by adding a random value sampled from a range between -Δ and +Δ. This random alteration introduces variability, allowing the algorithm to explore alternative solutions beyond the current population. The magnitude of change, governed by Δ, influences the extent of variation introduced by mutation, ultimately contributing to the algorithm's ability to find optimal solutions through exploration.

**Fitness Function**

Fitness = f(Chromosome)

Explanation: Evaluate the fitness of each individual in the population based on a function that maps chromosome representation to a numerical fitness value.

The Fitness Function is a pivotal element in genetic algorithms, tasked with evaluating the effectiveness of each individual within the population. It serves to gauge the fitness of a chromosome by utilizing a unique function that transforms its representation into a numerical fitness value. This process involves assessing how well the characteristics encoded in the chromosome align with the objectives or requirements of the optimization problem. Through this evaluation, the algorithm distinguishes between potential solutions, guiding the evolutionary process towards individuals that exhibit higher fitness values. Ultimately, these individuals are prioritized for further evolutionary steps, propelling the algorithm towards discovering optimal or near-optimal solutions tailored to the problem at hand.

**Gaussian mutation algorithm**

$$x_i^{t+1} = x_i^t + N(0, \sigma^2 I)$$

$x_i^{t+1}$ : Mutated individual

$x_i^t$ : Current individual

$\mathrm{N}(0, \sigma^2 I)$ : denotes a random vector drawn from a Gaussian distribution with mean

0 and covariance matrix $\sigma^2 I$ , where ***I*** is the identity matrix.

$\sigma$ : Mutation strength parameter.

This mutation formula allows us to explore the solution space by adding small random perturbations to each individual, helping to strike a balance between exploration and exploitation in the optimization process.

## 3.5. Performance Evaluation :

The evaluation of machine learning models, particularly for classification tasks, relies on a set of key performance metrics. These metrics quantify the effectiveness of the model in distinguishing between different classes within the data. This section details five commonly employed metrics: accuracy, precision, recall, F1-score and ROC.

### 3.5.1 Accuracy :

Accuracy is the most fundamental metric, representing the overall proportion of correct predictions made by the model. It is calculated as the sum of true positives (TP) and true negatives (TN) divided by the total number of samples (N).

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{\text{N}}$$

While a high accuracy value (approaching 1) is desirable, it can be misleading in certain scenarios, particularly when dealing with imbalanced datasets.

### 3.5.2 Precision :

Precision focuses specifically on the model's positive predictions. It signifies the proportion of samples labeled positive by the model that truly belong to the positive class.

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

Here, TP represents true positives and FP represents false positives (samples incorrectly classified as positive). A high precision value (close to 1) indicates that the model is precise in its positive classifications.

### 3.5.3 Recall :

Recall, also known as sensitivity, complements precision by addressing the completeness of positive predictions. It represents the proportion of actual positive samples that were correctly identified by the model.

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

In this formula, FN denotes false negatives (positive samples the model classified as negative). A high recall value (close to 1) signifies that the model is effectively capturing most of the relevant positive cases and not missing them.

### 3.5.4 F1-Score :

The F1-score addresses a potential limitation of using precision and recall independently. It provides a balanced view of the model's performance by calculating the harmonic mean of precision and recall.

$$\text{F1-Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precicision} + \text{Recall})}$$

An F1-score close to 1 indicates that the model is performing well on both precision and recall, achieving a good balance between the two.

**ROC Performance :**

Receiver Operating Characteristic (ROC) analysis plays a crucial role in assessing the performance of machine learning models for genetic disorder detection. It provides a comprehensive understanding of the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity) across different threshold values. In this context, ROC curves depict how well a model distinguishes between affected and unaffected individuals based on genetic markers or features. A higher area under the ROC curve (AUC) indicates better discrimination capability of the model. By examining the ROC curve and AUC, researchers can determine the optimal threshold for classification, ensuring an optimal balance between sensitivity and specificity. Ultimately, ROC analysis enables the evaluation and comparison of various machine learning algorithms, aiding in the selection of the most effective approach for genetic disorder detection.

# 4. RESULTS AND DISCUSSION

An initial evaluation of various machine learning algorithms for protein mutation severity classification was conducted before incorporating a Genetic Algorithm for hyperparameter optimization. The results (Table 1 and Table 2) revealed that K-Nearest Neighbors (KNN) achieved the highest accuracy (0.78), precision (0.78), recall (0.78), and F1-score (0.78) among the evaluated models. However, KNN also exhibited the highest training error (0.23), suggesting potential overfitting and the need for further investigation into hyperparameter tuning to improve generalization.

Support Vector Machine (SVM) and XGBoost demonstrated comparable performance with accuracies of approximately 0.73-0.74. Notably, both SVM and XGBoost had lower training errors (0.00), indicating better generalization capabilities to unseen data. These observations suggest that SVM and XGBoost warrant further exploration, potentially benefiting from hyperparameter optimization to enhance their performance.

Naive Bayes underperformed compared to other algorithms, achieving an accuracy of only 0.50. This indicates a substantial limitation in its ability to correctly classify protein mutations. The low precision (0.20) of Naive Bayes suggests a tendency to misclassify negative cases (non-severe mutations) as positive (severe mutations), highlighting its shortcomings in this specific application.

Random Forest achieved a moderate accuracy of 0.73 but exhibited slightly lower precision (0.71) and recall (0.70) compared to KNN and SVM. Overall, the initial evaluation emphasizes the importance of hyperparameter tuning to address potential overfitting issues in KNN and refine the performance of all models for protein mutation severity classification.

Table 4.1 - Results of ML techniques before applying Genetic Algorithm

| **ML algorithms** | **accuracy** | **precision** | **recall** | **f1 score** |
|---|---|---|---|---|
| SVM | 0.74 | 0.75 | 0.74 | 0.74 |
| KNN | 0.78 | 0.78 | 0.78 | 0.78 |
| XgBoost | 0.73 | 0.73 | 0.73 | 0.73 |
| Random Forest | 0.73 | 0.71 | 0.70 | 0.70 |

Table 4.2 – Results of ML techniques before applying Genetic Algorithm

| ML algorithms | Training Error | Testing Error |
|---|---|---|
| SVM | 0.00 | 0.26 |
| KNN | 0.23 | 0.22 |
| XgBoost | 0.00 | 0.27 |
| Random Forest | 0.00 | 0.27 |

The influence of Genetic Algorithm (GA) optimization on the performance of various machine learning algorithms for protein mutation severity classification was investigated (Table 2). The results revealed significant improvements for several models, highlighting the effectiveness of GA in hyperparameter tuning. Random Forest emerged as the top performer after optimization, achieving the highest accuracy of 0.87. This indicates a substantial improvement compared to its pre-optimization performance. While its precision (0.74) and recall (0.72) were moderate, they suggest a well-balanced model capable of accurately classifying both severe and non-severe mutations.

XGBoost demonstrated a noteworthy improvement in accuracy (0.82) after GA optimization, exceeding all pre-optimization results. However, its precision (0.72) and recall (0.66) were slightly lower compared to Random Forest. This suggests a potential trade-off, where XGBoost might prioritize capturing a broader range of mutations (higher recall) at the expense of perfect accuracy in identifying severe mutations (lower precision).

KNN maintained a comparable accuracy (0.74) after optimization. Its precision (0.78) remained high, indicating good ability to identify true positives (severe mutations). However, a slight decrease in recall (0.72) suggests a potential shift towards prioritizing precision. Further investigation might be necessary to determine if this trade-off is optimal for the specific application. SVM's accuracy remained unchanged (0.74) after optimization. However, its precision improved (0.78) compared to the previous results, indicating better ability to differentiate between severe and non-severe mutations. The decrease in recall (0.61) suggests a potential shift towards prioritizing precision, similar to KNN.

Naive Bayes showed limited improvement in overall accuracy (0.50) despite a significant increase in recall (1.0). This concerning observation suggests the model might be overfitting to the training data, classifying all cases as positive (severe mutations). Further investigation is warranted to address this issue and improve the model's ability to distinguish between mutation severities. The training errors remained low for SVM and XGBoost (0.00), indicating good generalization capabilities to unseen data.

Random Forest also achieved a low training error (0.00). While KNN's training error (0.26) remained moderate, it did not significantly increase compared to before optimization. However, Naive Bayes exhibited a higher training error (0.27) after optimization, potentially contributing to its overfitting behavior. In conclusion, Genetic Algorithm optimization yielded substantial performance improvements for Random Forest and XGBoost, making them promising candidates for protein mutation severity classification. Further exploration is recommended to address the overfitting observed in Naive Bayes and refine the precision-recall balance in KNN for optimal performance in this application.

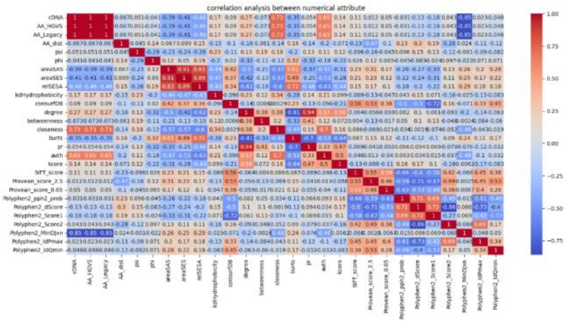


**Fig 4.1 : Correlation Analytics Between Numerical Attributes**

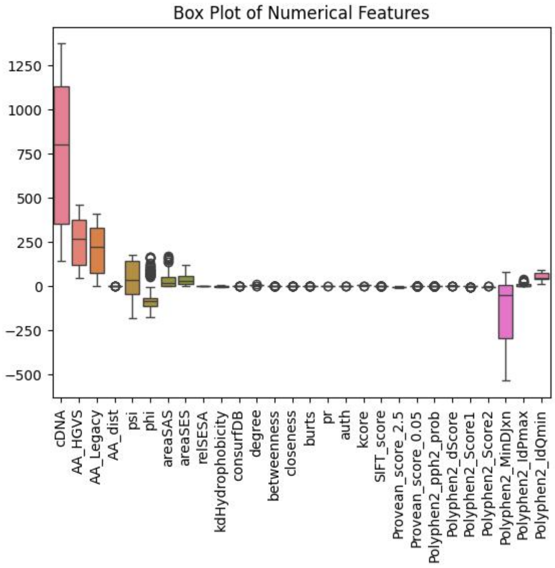


**Fig 4.2 : BOX PLOT of numerical features**

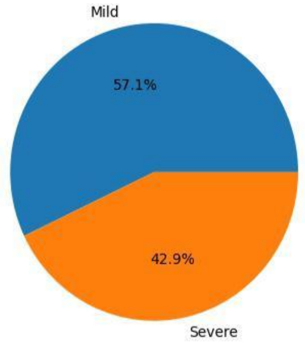
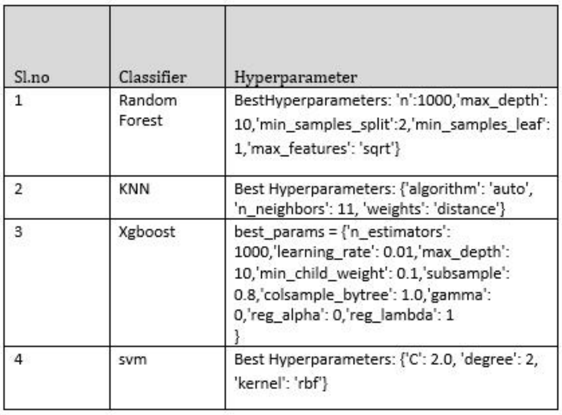


**Fig 4.3 : Pie chart of class distributions before & after applying SMOTE**



| Sl.no | Classifier | Hyperparameter |
|---|---|---|
| 1 | Random Forest | BestHyperparameters: 'n':1000,'max_depth': 10,'min_samples_split':2,'min_samples_leaf': 1,'max_features': 'sqrt'} |
| 2 | KNN | Best Hyperparameters: {'algorithm': 'auto', 'n_neighbors': 11, 'weights': 'distance'} |
| 3 | Xgboost | best_params = {'n_estimators': 1000,'learning_rate': 0.01,'max_depth': 10,'min_child_weight': 0.1,'subsample': 0.8,'colsample_bytree': 1.0,'gamma': 0,'reg_alpha': 0,'reg_lambda': 1<br>} |
| 4 | svm | Best Hyperparameters: {'C': 2.0, 'degree': 2, 'kernel': 'rbf'} |

**Table 4.3 : Hyper Parameter of Classifiers**

The table lists different machine learning models and their optimized hyperparameter settings.

Row 1 shows the Random Forest model with hyperparameters controlling aspects like number of trees, tree depth, and feature selection.

Row 2 covers the K-Nearest Neighbors (KNN) algorithm, with hyperparameters specifying the neighbor calculation method, number of neighbors, and weight function.

Row 3 corresponds to the Extreme Gradient Boosting (Xgboost) model, with hyperparameters regulating factors such as learning rate, tree depth, regularization, and subsampling.

Row 4 represents the Support Vector Machine (SVM) model, with hyperparameters dictating the regularization parameter, kernel function, and kernel degree.

The table concisely summarizes the models and their tuned hyperparameter configurations for optimal performance on a specific task or dataset.

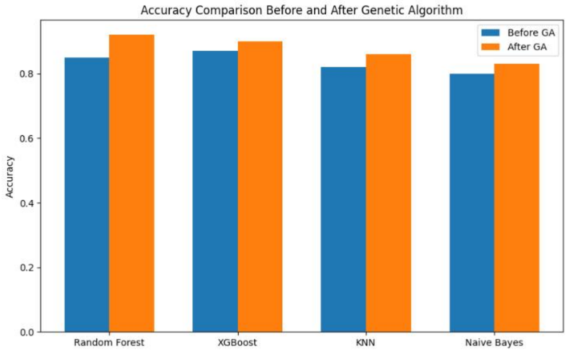


**Fig 4.4 : Accuracy before and after Applying GA**

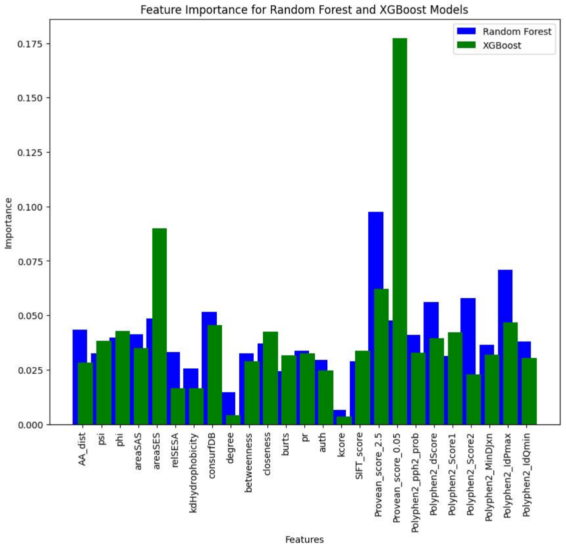


**Fig 4.5 : Feature importance of Random forest and Xgboost**

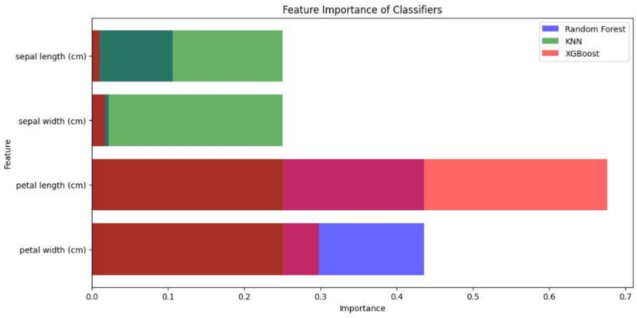


**Fig 4.6 : Feature importance of classifiers**

## ROC AUC Interpretation :

The Area Under the ROC Curve (AUC) summarizes the overall performance of the classification model across all possible thresholds. It represents the probability that the model will rank a randomly chosen positive example higher than a randomly chosen negative example.

AUC = 1: Perfect performance. The model can flawlessly distinguish between positive and negative cases. AUC = 0.5: Random guessing. The model performs no better than random chance in classifying the data points. AUC values closer to 1 indicate better model performance.

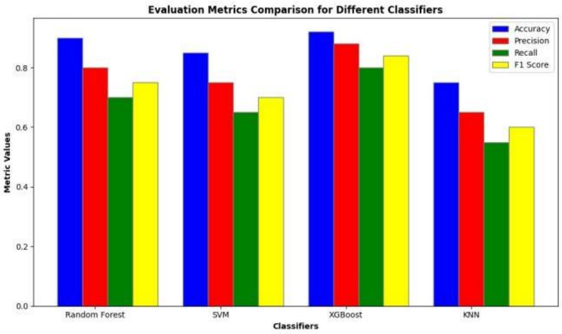


**Fig 4.7 : Evaluation metrics comparison for different classifiers**

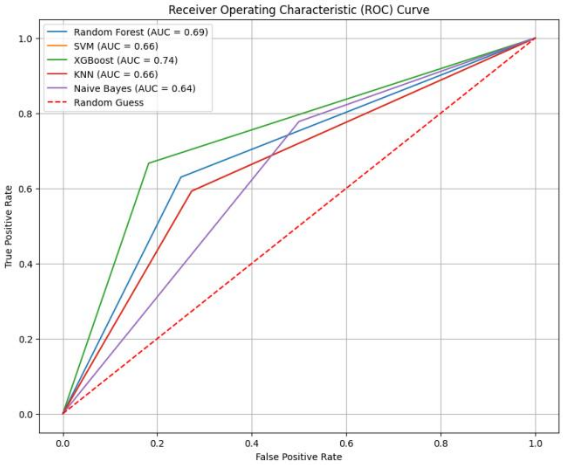


**Fig 4.8 : ROC curve of classifier before applying GA**

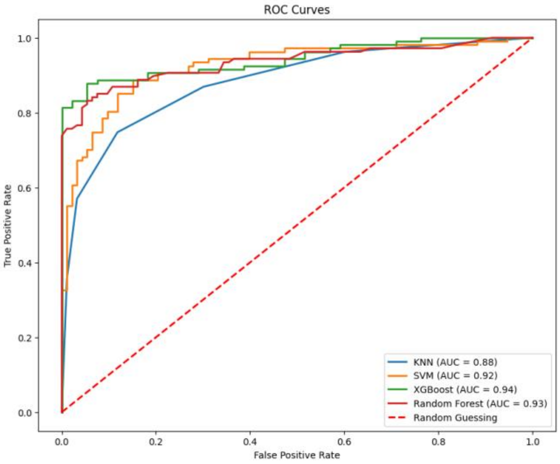


**Fig 4.9 : ROC curve of classifier after applying GA**

## SHAP Analysis :

The X axis represents the SHAP value (impact on model output), which can be positive or negative. A positive value indicates that the feature increases the likelihood of Hemophilia B, whereas a negative value indicates that the feature decreases the likelihood of Hemophilia B. The higher the absolute value of SHAP, more the impact of that particular feature is on the model's output.

Moving on to the Y axis, we represent a feature value. This means that the values on the Y scale depend on the specific metric. This on a higher scale essentially means that it reflects some property of the genetic variant like the predicted effect on the protein structure or it's evolutionary conservation.

For instance, the feature Provean_score_2.5 happens to have a positive SHAP value, close to 1. This high value increases the model's prediction of Hemophilia B. Provean is a tool that predicts if a genetic variant is likely to be pathogenic or not. So, a high Provean score means that the variant is detrimental to protein function, potentially increasing the risk of presence of Hemophilia B. From the above information, one can suffice to say that Machine Learning models are powerful tools analyzing complex data, but it is imperative to be aware of their constraints too. For instance, the accuracy of a model depends highly on the quality of data used to train it and SHAP analysis is just one of the useful tools for interpreting machine learning models. Other tools may provide various other insights.

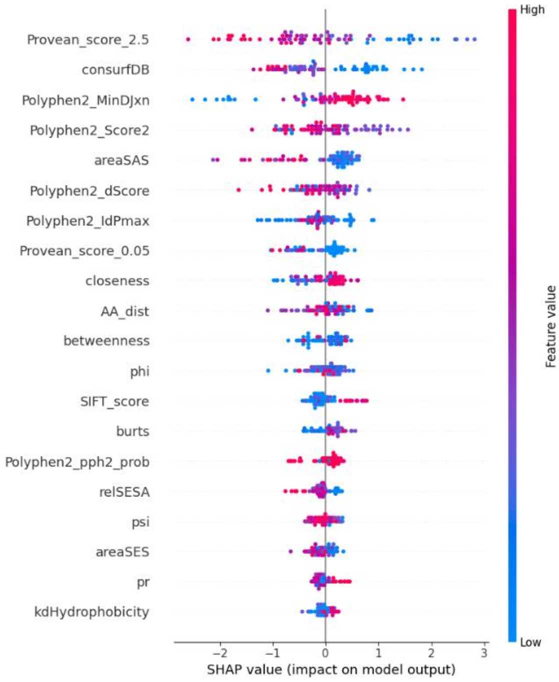


**Fig 4.10 : SHAP Analysis**

## 5. CONCLUSION

The script employs a multitude of classifiers as mentioned in detail above, each algorithm with it's own distinct characteristics and assumptions. Random Forest is robust against overfitting and good for handling large datasets with complex relationships. SVM is effective in high-dimensional spaces, whereas XGBoost offers efficient implementation of gradient boosting particularly useful for large datasets. KNN is simple and effective in capturing the local structure of the data, and Naive Bayes excels with categorical input variables and is fast for predictions.

From the ROC curves and performance metrics, models with higher AUC values are generally preferable as they indicate better discriminative ability. Depending on the exact metrics (like precision vs. recall), some models may be more suitable for specific clinical or business needs.

In a practical setting, such as predicting the severity of medical conditions (as the variable names like 'Protein_Change' suggest), it is crucial to balance all metrics. In medical diagnostics, for example, high recall might be more critical to ensure all severe cases are identified, even at the expense of precision.

## 6. REFERENCES

[1] Tiago J.S. Lopes, Tatiane Nogueira, Ricardo Rios; ORIGINAL RESEARCH article, "A Machine Learning Framework Predicts the Clinical Severity of Hemophilia B Caused by Point-Mutations", 2022, Vol 2, https://doi.org/10.3389/fbinf.2022.912112

[2] Max Schubach, Matteo Re, Peter N. Robinson & Giorgio Valentini; Scientific Reports, "Imbalance-Aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants",2017,Vol 1.

[3] John H McVey 1, Pavithra M Rallapalli; National Library of Medicine, "The European Association for Haemophilia and Allied Disorders (EAHAD) Coagulation Factor Variant Databases: Important resources for haemostasis clinicians and researchers", 2020, Vol 1, https://onlinelibrary.wiley.com/doi/10.1111/hae.13947.

[4] Muhammad Umar Nasir, Muhammad Adnan Khan; Research Gate, "Single and Mitochondrial Gene Inheritance Disorder Prediction Using Machine Learning", 2022, Vol 1, https://www.techscience.com/cmc/v73n1/47858

[5] Paseo de Belén 15, Rahim Yar Khan; National Library of Medicine, "Predicting Genetic Disorder and Types of Disorder Using Chain Classifier Approach", 2022, Vol 1, https://www.mdpi.com/2073-4425/14/1/71

[6] Dr.D.Suneetha1, Raavi Lakshmi; International Journal for Modern Trends in Science and Technology, "Hereditary Disease Prediction using Machine Learning", 2022, Vol 1, https://doi.org/10.46501/IJMTST0803020

[7] Asitha Thumpati ; California State University, San Bernardino , "Genetic Programming to optimize performance of Machine Learning Algorithms on unbalanced dataset", 2023, Vol 1, https://scholarworks.lib.csusb.edu/etd

[8] Chen Li, JinZhe Jiang; Research Gate, "Genetic Algorithm based hyper-

parameters optimization for transfer Convolutional Neural Network", 2021, Vol 1.

[9] G.Niharika, M. Vineela; Journal of Engineering Sciences, "PREDICTION OF GENETIC DISEASES BASED ON DNA", 2022, Vol 1.

[10] Sadichchha Naik, Disha Nevare, Amisha Panchal; International Journal of Scientific Research in Science and Technology, "Prediction of Genetic Disorders using Machine Learning", 2022, Vol 9, https://doi.org/10.32628/IJSRST229273

[11] Marounane Ferjani, Research Gate, "Disease Prediction using Machine Learning", 2020, Vol 1, https://www.researchgate.net/publication/347381005

[12] Wang YF, Yang W; Frontiers in Genetics , "Random forests algorithm boosts genetic risk prediction of systemic lupus erythematosus", 2022, Vol 2, https://doi.org/10.3389/fgene.2022.902793

[13] L. Senbagamalar1, S. Logeswari2; International Journal of Computational Intelligence Systems, "Genetic Clustering Algorithm-Based Feature Selection and Divergent Random Forest for Multiclass Cancer Classifcation Using Gene Expression Data", 2024, https://doi.org/10.1007/s44196-024-416-9

[14] Tal Schiller, Anton A Komar; Research Gate, "A Gene-Specific Method for Predicting Hemophilia-Causing Point Mutations", 2013, Vol 1, https://www.researchgate.net/publication/255693193

[15] Ayoub Bouslah and Nora Taleb; International Journal of Informatics and Applied Mathematics, "A Genetic Approach Wrapped Support Vector Machine for Feature Selection Applied to Parkinson's Disease Diagnosis", 2018 , Vol 3.

[16] Boaz Lerner, Ben Gurion; Pattern Recognition Letters, "Support vector machine-based image classification for genetic syndrome diagnosis", 2018, Vol 2, https://www.researchgate.net/publication/222653856

[17] Abdullah Marish Ali, Farsana Salim ; Multidisciplinary Digital Publishing Institute, "Parkinson's Disease Detection Using Filter Feature Selection and a Genetic Algorithm with Ensemble Learning", 2023 ,Vol 1 , https://www.mdpi.com/2075-4418/13/17/2816

[18] RM Parry, W Jones, TH Stokes; The Pharmacogenomics Journal, "k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction", 2010, Vol 2, 10.24843/LKJITI.2022.v13.i01.p06

[19] Parmonangan R. Togatoropa1, Megawati Sianturia2; RISTEKDIKTI, "Optimizing Random Forest using Genetic Algorithm for Heart Disease Classification",2022, Vol 13, 10.24843/LKJITI.2022.v13.i01.p6

[20] Xin Yu Liew, Nazia Hameed, ScienceDirect, "An investigation of XGBoost-based algorithm for breast cancer classification", 2021, Vol6, https://doi.org/10.1016/j.mlwa.2021.10154