e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

DISEASE PREDICTION USING GENETIC DATA

Mrs. D. Ashwani[*1], Padakanti Pranathi[*2], Koppurapu Harshavardhan Reddy[*3], Desharaju Sai Anupam[*4], Vennavelly Pradeep Reddy[*5]

[*1]Internal Guide, ACE Engineering College Of Computer Science And Engineering, Ghatkesar, Telangana, India.

[*2,3,4,5]Student, ACE Engineering College Of Computer Science And Engineering, Ghatkesar, Telangana, India.

DOI: https://www.doi.org/10.56726/IRJMETS67124

ABSTRACT

One of the leading fields for the implementation of machine learning is healthcare. Most medical facilities and research institutes strive to be advanced so that they can make better decisions for patient diagnosis and care. Machine learning prediction models enable us to process large volumes of complex medical datasets and provide better predictions for disease diagnosis. Identifying various critical and terminal diseases allows medical professionals to start treatment at an earlier stage. Any form of disease prediction significantly increases treatment progress, potentially avoiding the terminal stage or even helping identify potential outbreaks. This paper focuses on working with a genetic dataset obtained from a gene microarray to better understand how different classifiers can be applied and to compare results.

I. INTRODUCTION

Disease prediction using genetic data has emerged as a promising field in personalized medicine, offering the potential to predict, prevent, and treat various diseases more effectively. Our genetic makeup—the DNA we inherit from our parents—plays a significant role in determining our susceptibility to a wide range of diseases, including both common conditions such as heart disease, diabetes, and cancer, as well as rare genetic disorders. With advancements in genomics and biotechnology, it is now possible to sequence an individual's genome and analyze genetic variations that may contribute to disease risk. This opens new possibilities in predictive medicine, where genetic data can estimate the likelihood of developing certain diseases before symptoms appear.

Key Aspects of Disease Prediction Using Genetic Data:

1. **Genetic Variations** – Single nucleotide polymorphisms (SNPs) influence disease risk.
2. **Genomic Technologies** – Whole Genome Sequencing (WGS) and Genome-Wide Association Studies (GWAS) provide insights into disease-related genes.
3. **Machine Learning and AI** – Advanced computational models identify patterns and predict disease risks.
4. **Personalized Medicine** – Tailored treatment plans based on genetic profiles.
5. **Ethical Considerations** – Privacy, data security, and discrimination concerns need to be addressed.

II. PROBLEM STATEMENT

Challenges:

The growing volume of genetic data presents an opportunity for disease prediction, but current medical systems struggle with:

- Efficiently analyzing and interpreting genetic sequences.
- Ethical concerns regarding data privacy and genetic discrimination.

III. PROPOSED SYSTEM

The proposed system aims to improve disease prediction by leveraging machine learning techniques on genetic data. Key features include:

- **Data Collection:** Uses gene microarray datasets containing genetic variations.
- **Data Processing:** Cleans and preprocesses genetic sequences.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

- **Feature Selection:** Identifies key genetic markers linked to specific diseases.
- **Machine Learning Model:** Trains predictive models (e.g., Random Forest, Neural Networks) to analyze genetic data.
- **User Interface:** Provides a web-based platform where users can upload genetic data and receive disease predictions.

IV. HARDWARE REQUIRMENTS

Implement and deploy the proposed system, the following hardware specifications are required:

- **Processor:** Minimum Quad-core (Intel i5 or equivalent)
- **RAM:** At least 8GB (16GB recommended for large datasets)
- **Storage:** SSD with at least 256GB free space
- **GPU:** Recommended for high-performance computing (e.g., NVIDIA GTX 1050 or higher)

V. MODEL ANALYSIS

The effectiveness of a disease prediction system using genetic data heavily depends on the selection of machine learning models and their performance evaluation. This section provides an analysis of different models used in the project, their performance metrics, and a comparison with existing approaches.

1. Random Forest Classifier

a. An ensemble learning technique that uses multiple decision trees to improve accuracy.

b. Handles large feature sets well and reduces overfitting through averaging predictions.

2. Extra Trees Classifier

a. Extra Trees Classifier is an ensemble of decorrelated decision trees.

b. At each node, it selects k random features and splits at a random threshold.

3. Neural Networks (Deep Learning)

a. Uses layers of artificial neurons to identify complex patterns in genetic sequences.

b. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can be applied to analyze gene expression and sequence data.

4. K-Nearest Neighbors (KNN)

a. A simple distance-based model that classifies new samples based on similarity to existing data points.

b. Works best for smaller datasets but struggles with high-dimensional genetic data.

5. Naïve Bayes Classifier

a. A probabilistic model that assumes feature independence.

b. Fast but less effective for complex genetic interactions.

2. Performance Evaluation Metrics

The models were evaluated based on several key performance indicators:

- **Accuracy** – Measures the overall correctness of predictions.
- **Precision** – Determines how many of the predicted positive cases are actually correct.
- Recall (Sensitivity) – Indicates how well the model identifies actual positive cases.
- **F1-Score** – A harmonic mean of precision and recall, useful for imbalanced datasets.
- **ROC-AUC Score** – Evaluates the ability to distinguish between disease and non-disease cases.

3. Insights & Improvements

- **Feature Selection:** Selecting top-ranking genetic markers improved model accuracy.
- **Hyperparameter Tuning:** Optimizing parameters (e.g., number of trees in Random Forest, kernel function in SVM) enhanced performance.
- **Data Augmentation:** Increasing the training data size further improved deep learning model results.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

- **Hybrid Models:** Combining Random Forest with Neural Networks could yield better accuracy by leveraging both interpretability and deep learning power.

VI. IMPLEMENTATION

Importing Libraries

In [1]:
```
import pandas as pd
from sklearn import preprocessing
import matplotlib.pyplot as plt
import logging
logging.getLogger("sklearn").setLevel(logging.ERROR)

import warnings
warnings.filterwarnings('ignore')
```

Importing Libraries

In [1]:
```
import pandas as pd
from sklearn import preprocessing
import matplotlib.pyplot as plt
import logging
logging.getLogger("sklearn").setLevel(logging.ERROR)

import warnings
warnings.filterwarnings('ignore')
```

Data Loading

Data is loaded from the file system and the sample labels are encoded to numerical values using the label encoder.

In [2]:
```
train_df = pd.read_csv('pp5i_train.gr.csv')
test_df = pd.read_csv('pp5i_test.gr.csv')
class_df = pd.read_csv('pp5i_train_class.txt')

class_np = class_df.to_numpy()

le = preprocessing.LabelEncoder()
le.fit(class_np)
train_class = le.transform(class_np)
```

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

Data Loading

Data is loaded from the file system and the sample labels are encoded to numerical values using the label encoder.

In [2]:
```
train_df = pd.read_csv('pp5i_train.gr.csv')
test_df = pd.read_csv('pp5i_test.gr.csv')
class_df = pd.read_csv('pp5i_train_class.txt')

class_np = class_df.to_numpy()

le = preprocessing.LabelEncoder()
le.fit(class_np)
train_class = le.transform(class_np)
```

Data Loading

Data is loaded from the file system and the sample labels are encoded to numerical values using the label encoder.

In [2]:
```
train_df = pd.read_csv('pp5i_train.gr.csv')
test_df = pd.read_csv('pp5i_test.gr.csv')
class_df = pd.read_csv('pp5i_train_class.txt')

class_np = class_df.to_numpy()

le = preprocessing.LabelEncoder()
le.fit(class_np)
train_class = le.transform(class_np)
```

Data Loading

Data is loaded from the file system and the sample labels are encoded to numerical values using the label encoder.

In [2]:
```
train_df = pd.read_csv('pp5i_train.gr.csv')
test_df = pd.read_csv('pp5i_test.gr.csv')
class_df = pd.read_csv('pp5i_train_class.txt')

class_np = class_df.to_numpy()

le = preprocessing.LabelEncoder()
le.fit(class_np)
train_class = le.transform(class_np)
```

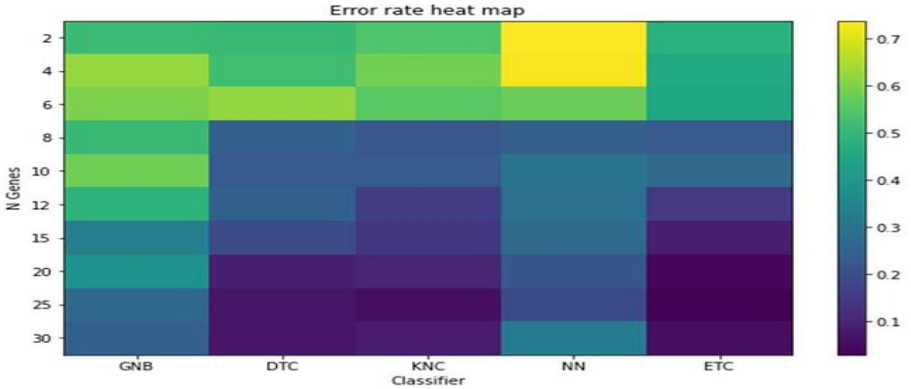
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

[9]:
```
filename="pp5i_train.bestN.csv"
data_arr = np.genfromtxt(filename,delimiter=',')
x_trainNT = data_arr
y_trainNT = best_genes_cls

clf = funcdict[maxCV]()

if maxCV=='KNeighborsClassifier':
  clf = funcdict[C](3)
elif maxCV=='ExtraTreesClassifier':
  clf = funcdict[C](n_estimators=350)
elif maxCV=='MLP':
  clf = MLPClassifier(activation = 'relu', solver = 'sgd', hidden_layer_sizes= (25, 25),random_state = 1, max_iter=250)
else:
  clf = funcdict[C]()

clf.fit(x_trainNT,y_trainNT)

from sklearn.model_selection import cross_val_score
scores = cross_val_score(clf, x_trainNT, y_trainNT,cv=5)

print("Best N         : ",maxNV)
print("Best Clasifier : ",maxCV)
print("Best Accuracy  : ",np.mean(scores))
```

```
Best N         :  25
Best Clasifier :  ExtraTreesClassifier
Best Accuracy  :  0.9714285714285715
```

[3]:
```
ttdf_sno=train_df['SNO']
ttdf_rem=train_df.iloc[:,1:]
ttdf_rem=ttdf_rem.clip(20,16000)

tsdf_sno=test_df['SNO']
tsdf_rem=test_df.iloc[:,1:]
tsdf_rem=tsdf_rem.clip(20,16000)


ttdf_cal = ttdf_rem.max(axis=1)/ttdf_rem.min(axis=1)
ttdf_cal = abs(ttdf_cal)
del_ind = ttdf_cal[ttdf_cal<2].index

train_tdf = pd.concat([ttdf_sno.drop(del_ind),ttdf_rem.drop(del_ind)],axis=1,sort=False)
test_tdf = pd.concat ([tsdf_sno.drop(del_ind),tsdf_rem.drop(del_ind)],axis=1,sort=False)

from sklearn.feature_selection import f_classif
tTrain_tdf = train_tdf.T[1:]
new_train = f_classif(tTrain_tdf,train_class)
train_tdf['rank']=new_train[0]
test_tdf['rank']=new_train[0]

train_tdf=train_tdf.sort_values('rank',ascending=False)
test_tdf=test_tdf.sort_values('rank',ascending=False)

print("\033[4m Training Data: \033[0m \n",train_tdf,"\n")
print("\033[4m Testing Data: \033[0m \n",test_tdf,"\n")

training_data = train_tdf.drop('SNO',axis=1)
training_data = training_data.drop('rank',axis=1)
training_data = training_data.to_numpy()

p=plt.plot(training_data)
```

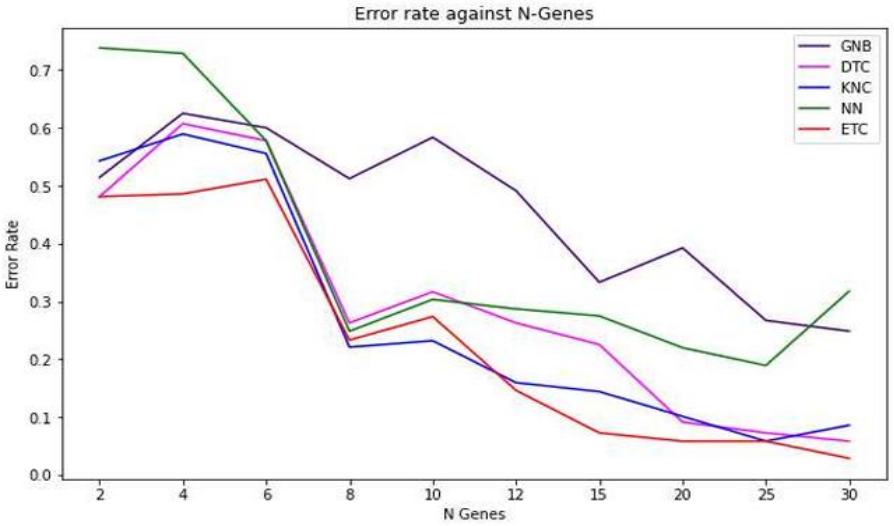
VII. OUTPUT



As illustrated in the graph, the error rate of the classifiers tends to reduce on the higher N-genes values.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:02/February-2025 Impact Factor- 8.187 www.irjmets.com

Among the classifiers trained, the Extra tree classifier predict the disease class with best accuracy rate of 97.14% for the subset of top gene samples with N value of 25.



This graph illustrates the relationship between the error rate and the number of genes (N-Genes) used in various classification models. The x-axis represents the number of genes included in the analysis, while the y-axis shows the error rate of predictions, with lower values indicating better performance. Different models, including Gaussian Naive Bayes (GNB), Decision Tree Classifier (DTC), K-Nearest Neighbors Classifier (KNC), Neural Network (NN), and Extra Trees Classifier (ETC), are represented by distinct colored lines. Initially, when fewer genes are used, the error rates are relatively high for all models. As the number of genes increases, there is a significant reduction in error rates, particularly between 8 and 15 genes, where most models show their best improvement. At higher numbers of genes, the error rates stabilize, with ETC and NN achieving the lowest error rates overall, indicating superior performance. Conversely, some models, like GNB and DTC, show fluctuating trends, suggesting a sensitivity to the number of genes included. This graph highlights how the incorporation of additional genetic features can improve model accuracy, though the degree of improvement varies across algorithm.

VIII. CONCLUSION

In this project, we developed and compared several machine learning classifiers for predicting disease using dataset collected from gene microarray. The classifiers are trained in the labelled training gene samples and predicted on the provided unlabeled test sample. The most efficient classifier among them was identified as Extra Tree Classifier with best accuracy rate. Based on the proposed classification model, the disease prediction can be done for any sample collected over the microarray and the patient can be diagnosed in a most efficient manner. As a future work to this project, we can assess the classifiers on more datasets and disease classes, examining its efficiency on predicting the disease on more complicated gene datasets.

IX. REFERENCES

[1] J.R. Quinlan, "Decision Trees As Probabilistic Classifiers," Proceedings of the Fourth International Workshop on Machine Learning.

[2] F. Xing, L. Yang, "Machine Learning and Its Application in Microscopic Image Analysis," Machine Learning and Medical Imaging, 2016.

[3] Pierre Geurts, Damien Erns, Louis Wehenkel, "Extremely Randomized Trees," Springer Science.

[4] Hongmei Yan, Yiangtao Jiang, et al., "A Multilayer Perceptron-Based Medical Decision Support System for Heart Disease Diagnosis."

[5] Shadab Adam Pattekari, Asma Parveen, "Prediction System for Heart Disease Using Naive Bayes," International Journal of Advanced Computer and Mathematical Sciences, pp. 290-294.

[6] Visscher et al., "10 Years of GWAS Discovery: Biology, Function, and Translation," The American Journal of Human Genetics, 2017.

[7] F.S. Collins, H. Varmus, "A New Initiative on Precision Medicine," The New England Journal of Medicine, 2015.

[8] N. Mullins et al., "Polygenic Risk Scores for Psychiatric Disorders and Their Implications for Public Health," The Lancet Psychiatry, 2020.

[9] L.A. Hindorff et al., "The Next Generation of GWAS and the Potential for Precision Medicine," Nature Reviews Genetics, 2018.

[10] E. Zeggini, L.J. Scott, "Genome-wide Association Studies in Complex Diseases," Nature Reviews Genetics, 2020.

[11] J.C. Denny et al., "The eMERGE Network: A Consortium of Biorepositories and Electronic Medical Records to Study the Genetic Basis of Disease," Journal of the American Medical Informatics Association, 2013.

[12] W.K. Chung, E. McPherson, "Clinical Applications of Whole Genome Sequencing," JAMA, 2017.

[13] N. Pashayan et al., "Stratified Screening for Breast Cancer Using Polygenic Risk Scores and Mammographic Density: A Population-based Simulation Study," The Lancet Oncology, 2019.

[14] N. Shah, W.G. Hill, "Machine Learning Approaches for Genetic Risk Prediction in Complex Diseases," Nature Reviews Genetics, 2021.

[15] Y. Jiang et al., "Genetic Data Integration with Machine Learning Models for Cancer Risk Prediction," Nature Communications, 202