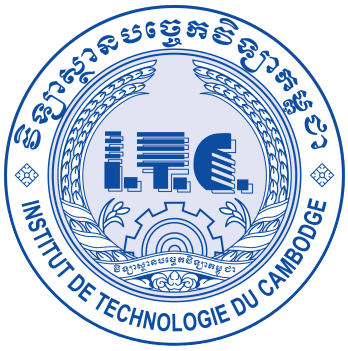


Disease Prediction Using Machine Learning

Mini Project
2023-2024

3rd year Engineer's Degree in Data Science
Department of Applied Mathematics and Statistics
Institute of Technology of Cambodia

Group members:
HUON SITHAI (ID: e20210954)
BUN RATNATEPY (ID: e20210320)
CHHIN VISAL (ID: e20210742)
EN SREYTHOM (ID: e20210084)

Submission Date: June 21, 2024

Lecturers:
Mr. Nhim Malai

Contents

Abstract

Machine learning techniques have gained significant attention in the field of healthcare for predicting and diagnosing diseases. This abstract presents an overview of the application of machine learning algorithms for disease prediction. By leveraging large datasets and advanced computational models, machine learning algorithms can analyze patient data, identify patterns, and generate accurate predictions about the occurrence, progression, and outcomes of various diseases. The process begins with data collection, where diverse information such as medical records, genetic profiles, lifestyle data, and environmental factors are gathered. Feature selection and preprocessing techniques are applied to extract relevant and meaningful features from the collected data. These features are then used to train machine learning models, such as decision trees, support vector machines, random forests, or deep learning architectures, using supervised or unsupervised learning approaches. The trained models are evaluated using appropriate metrics and validated to ensure their reliability and performance. Once validated, the models can be deployed to predict diseases in real-time by inputting new patient data. The predictions generated by these models can aid healthcare professionals in early detection, risk assessment, personalized treatment planning, and disease management.Furthermore, the abstract discusses the challenges and limitations associated with disease prediction using machine learning, including data quality, interpretability, model generalization, and ethical considerations. It emphasizes the need for robust data privacy and security measures to protect sensitive patient information. Overall, the application of machine learning techniques in disease prediction has the potential to revolutionize healthcare by enabling accurate and timely diagnoses, improving patient outcomes, and optimizing resource allocation. However, ongoing research and collaboration between healthcare professionals, data scientists, and policymakers are essential to address the remaining challenges and ensure the responsible and effective implementation of machine learning in healthcare settings.

Key Words

Logistic Regression
Random Forest
Decision Trees
K-Nearest Neighbors
Naive bayes
Support Vector Machine

1 Introduction

1.1 Background and Motivations

This section sets the stage by exploring the context surrounding disease prediction and the advancements in machine learning technology driving interest in this area. It discusses the transformative impact of technology on healthcare, the increasing demand for accurate predictive models, and the challenges faced by individuals in accessing healthcare services. The motivation lies in leveraging machine learning to address these challenges and improve patient outcomes, diagnostic accuracy, and treatment strategies.

1.2 Research Objectives

In this section, the specific goals and aims of the literature review are outlined. It details the objectives, such as critically reviewing existing research on machine learning in disease prediction, identifying different methodologies and algorithms used, examining data sources and features, assessing advantages and limitations, and highlighting future research directions. These objectives provide a roadmap for the literature review and guide the analysis and synthesis of existing research.

1.3 Significance of This Work

Here, the importance and relevance of the literature review are emphasized. It discusses how synthesizing existing research contributes to advancing knowledge in healthcare analytics, informs decision-making processes for stakeholders, and guides the implementation of machine learning models in disease prediction and management. Additionally, it highlights how the review addresses gaps in the literature, offers insights into emerging trends and opportunities, and contributes to the ongoing dialogue in the field.

1.4 Dissertation Structure

This section provides an overview of the structure and organization of the dissertation without summarizing its content. It outlines the chapters or sections included, such as literature review, methodology, results, discussion, and conclusion. Additionally, it briefly describes the content of each chapter and explains how they contribute to fulfilling the research objectives outlined earlier.

2 Literature Reviews

2.1 First paper

- Title: Disease prediction using machine learning
- Author(s): Neha Gupta, Kriti Gandhi, Shafali Dhall
- Publication year: June 2020
- Link to article
- Method: Naive Bayes Classifier, Random Forest, Logistic Regression(LR), SVM, KNN, CART, LDA
- Comparative Analysis:

Table 1: Model Accuracy Comparison

Model	**Accuracy**
Logistic Regression	93.87
CART	96.3
KNN	96.01
Naïve Bayes	96.99
SVM	95.62
LDA	95.05
Random Forest	80.85

2.2 Second paper

- Title: Human Disease Prediction using Machine Learning Techniques and Real-life Parameters
- Author(s): K. Gaurav, A. Kumar, P. Singh, A. Kumari, M. Kasar*, T. Suryawanshi
- Publication year: 31 March 2023
- Link to article
- Method: Naive Bayes Classifier, Random Forest, Decision Tree, Logistic Regression(LR), SVM, KNN

Table 2: Comparative Analysis

Ref.	**Algorithm Used**	**Advantages**	**Limitation(s)**	**Accuracy**
[17]	Naive Bayes Classifier	Highly Scalable	Only for independent features it works accurately	94.8%
[18]	Random forest, Decision tree, Naive Bayes	Good accuracy for predicting disease	Model needs to be enhanced via ensemble model	90%
[15]	Weighted KNN	Smoother decision surface, less data dependency	Due to the issue of overfitting, model is not scalable	93.5%
[29]	SVM	Faster Execution, Less Space complexity	Not Suitable for Multi-parameter	76%
[30]	SVM	Faster Execution, Less Space complexity	Not Suitable for Multi-parameter	90%
[32]	Logistic Regression(LR)	It makes assumption about distribution	Over-Fitting issue is there. It requires less multi-collinearity	75%
Proposed Method	Random Forest	The dataset is suitable for Random Forest	Can be improved if time series dataset is provided	97%

2.3 Third paper

- Title: THE PREDICTION OF DISEASE USING MACHINE LEARNING
- Author(s): Dr. C k gomathy, Mr.A.Rohirt Naidu
- Publication year: 10 October 2021
- Link to article
- Method: Naive Bayes Classifier, Random Forest, Decision Tree, SVM, KNN

Model	**Accuracy** (%)
Decision Tree	84.5
Random Forest	98.95
Naïve Bayes	89.4
SVM	96.49
KNN	71.28

Table 3: Model Accuracies

2.4 Forth paper

- Title: Disease Prediction using Machine Learning Algorithms
- Author(s): Sneha Grampurohit, Chetan Sagarnal
- Publication year: 07 June 2020
- Link to article
- Method: Naive Bayes Classifier, Random Forest, SVM, KNN, LDA, Logistic Regression, CART

Algorithms	Accuracy Scores	Standard Deviation	No. Of Features
LOGISTIC REGRESSION	0.988790	0.003105	124
CART	0.963035	0.006168	61
KNN	0.960116	0.008465	52
NAÏVE BAYES	0.969986	0.004774	52
SVM	0.956250	0.006451	52
LDA	0.950578	0.006168	77
RANDOM FOREST	0.808566	0.042578	52

Table 4: Comparison of Different Algorithms

2.5 Fifth paper

- Title: DISEASE PREDICTION USING MACHINE LEARNING
- Author(s): V. SHARON ROSE
- Publication year: March 2021
- Link to article
- Method: Naive Bayes Classifier, Random Forest, Decision Tree

Model	Accuracy
Decision Tree	95.12
Random Forest	95.12
Naïve Bayes	95.12

Table 5: Model Accuracies

3 Methods and Materials

	count	mean	std	min	25%	50%	75%	max
itching	4920.0	0.137805	0.344730	0.0	0.0	0.0	0.0	1.0
skin_rash	4920.0	0.159756	0.366417	0.0	0.0	0.0	0.0	1.0
nodal_skin_eruptions	4920.0	0.021951	0.146539	0.0	0.0	0.0	0.0	1.0
continuous_sneezing	4920.0	0.045122	0.207593	0.0	0.0	0.0	0.0	1.0
shivering	4920.0	0.021951	0.146539	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...
small_dents_in_nails	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
inflammatory_nails	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
blister	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
red_sore_around_nose	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0
yellow_crust_ooze	4920.0	0.023171	0.150461	0.0	0.0	0.0	0.0	1.0

132 rows × 8 columns

3.1 Dataset

The dataset used in this study was sourced from Kaggle , comprising comprehensive medical records aimed at disease prediction. It includes (4962 rows and 133 columns) symptoms, diagnosis outcomes, and patient demographics, Each row represents a specific case or instance, while each column represents a symptom or feature. combined into a single dataset for analysis.

3.2 Exploratory Data Analysis

Exploratory Data Analysis is essential in machine learning as it helps in understanding, cleaning, preparing, and exploring the data before building models. It enables you to make informed decisions, improve the quality of the data, and increase the chances of building accurate and robust machine learning models.

There are Bar Plots and correlation to do Exploratory Data Analysis:

1. We use **Bar Plots** methods to do Exploratory Data Analysis. Bar plots provide a clear and concise way to summarize and visualize categorical data, making them valuable tools in Exploratory Data Analysis for predicting diseases. However, it's important to use them in conjunction with other statistical techniques and visualization methods to gain a comprehensive understanding of the data and draw meaningful conclusions.

2. We use **correlation plot** to do Exploratory Data Analysis to examine the relationships between variables in a dataset. They provide valuable insights into the strength and direction of the linear association between pairs of variables. it's important to note that correlation does not imply causation. While a strong correlation between two variables suggests an association, it does not necessarily mean that one variable causes the other. Therefore, additional research, such as controlled experiments or longitudinal studies, is often required to establish causal relationships between variables and disease outcomes.

3.3 Logistic Regression

1. The motivation to use logistic regression algorithm
Dataset is a disease so it is a type of categorical and logistic regression works on each category classification and it is commonly used to predict disease outcomes or classify individuals into disease categories based on a dataset. The logistic regression model uses a mathematical formula to predict the probability of an individual having a disease or not. The formula is derived from the logistic function, also known as the sigmoid function, which maps a linear combination of the features to a probability value between 0 and 1.
Logistic regression is suitable for binary classification problems, where the goal is to predict whether an individual has a disease or not. It provides probability estimates, allowing for assessing the likelihood of disease occurrence based on input variables. Logistic regression offers interpretability, with coefficients indicating the direction and magnitude of the variables' influence on disease prediction. It can handle both continuous and categorical variables, accommodating the diverse data types often encountered in disease prediction tasks. Logistic regression is a robust and relatively simple algorithm, making it accessible and efficient for disease prediction. It has a well-established methodology and has been extensively studied and validated in medical research. However, the choice of algorithm should consider the specific characteristics of the data, the nature of the problem, and the goals of the analysis.

2. Briefly about the logistic regression algorithm
Logistic regression is an algorithm used for binary classification tasks, specifically to predict whether an individual has a disease or not. It models the relationship between input variables (predictors) and the probability of belonging to a specific class (having the disease). The algorithm assumes a linear relationship between predictors and the log-odds of the outcome, transformed using the logistic function. The model estimates coefficients for each predictor, indicating their influence on disease prediction. The model is trained using labeled data, optimizing parameters to minimize a loss function and fit the predicted probabilities to observed outcomes. Predictions for new examples are made by calculating probabilities based on the trained model and applying a threshold for classification. Logistic regression has assumptions regarding linearity and independence of observations that should be considered. It is popular for disease prediction due to its ability to handle binary classification, provide probability estimates, offer interpretability, and its established use in medical research.

3. Formula of linear regression

Hypothesis:

$$h(x) = g(z) = \frac{1}{1 + e^{-z}}$$

where

$$Z = \sum_{j=0}^{n} (\theta_j x_j)$$

if $y = 1$, we want $h(x) \approx 1 \quad z \gg 0$
if $y = 0$, we want $h(x) \approx 0 \quad z \ll 0$

$\theta_0, \theta_1, ...$ are the coefficients or weights associated with each predictor variable.
$x_0, x_1, ...$ represent the predictor variables (features) related to the disease.

Cost Function

$$Cost = -y \log(h(x)) - (1 - y) \log(1 - h(x))$$

if $y = 1$, we want $z \gg 1$ (not just $\gg 0$)
if $y = 0$, we want $z \ll -1$ (not just $\ll 0$)

3.4 Decision Trees

1. Talk about motivation to use Decision Trees algorithm
Decision Trees algorithm is widely used for disease prediction due to its motivational factors.

- Interpretability: Decision trees provide a clear and understandable representation of the decision-making process, allowing easy interpretation of feature contributions to disease prediction.
- Feature Importance: Decision trees assign importance to features, helping identify the most influential predictors for disease prediction.
- Nonlinear Relationships: Decision trees can capture complex nonlinear relationships between predictors and disease outcomes, accommodating interactions, thresholds, and non-monotonic patterns.
- Handling Missing Values: Decision trees can handle missing values by creating surrogate splits based on available data, making them practical for real-world medical datasets.
- Scalability: Decision tree algorithms are computationally efficient and can handle large datasets with numerous predictors, making them suitable for medical applications with a high number of variables.
- Ensemble Methods: Decision trees can be combined through ensemble methods like random forests or gradient boosting, improving predictive accuracy and model robustness.
- Clinical Applicability: Decision trees provide actionable insights for clinical decision-making, aiding risk stratification, diagnosis, treatment selection, and prognosis assessment.

2. Talk briefly about Decision Trees algorithm
The decision trees algorithm is commonly used for predicting diseases due to its effectiveness and interpretability.

- Data Preparation: The algorithm requires a labeled dataset with input features (e.g., patient characteristics, medical history, test results) and corresponding disease outcomes.
- Tree Construction: The algorithm builds a tree-like structure by recursively partitioning the data based on feature values. It selects the most informative feature and splits the data into subsets that maximize the separation of diseased and non-diseased cases.
- Node Splitting: At each node of the tree, the algorithm determines the best feature and threshold to split the data, based on criteria such as information gain or Gini impurity. This process creates decision nodes representing conditions or rules.

- Recursive Process: The splitting process continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of samples required to split a node. This results in a tree structure where leaf nodes represent disease predictions.
- Prediction: To predict disease for a new sample, it traverses the decision tree based on the values of its features. The sample follows the decision path from the root to a leaf node, and the majority class of training samples in that leaf node determines the disease prediction.
- Interpretability: Decision trees offer interpretability, as the tree structure allows for easy understanding of how different features contribute to disease prediction. It provides insights into the most influential factors and their thresholds.
- Handling Missing Values: Decision trees can handle missing values by using surrogate splits or assigning probabilities to branches based on available data, making it practical for medical datasets with missing information.
- Ensemble Methods: Decision trees can be combined through ensemble methods like random forests or gradient boosting, which aggregate predictions from multiple trees to improve accuracy and robustness.

Present the formula of Decision Trees
The decision trees algorithm does not have a specific formula like a mathematical equation. Instead, it involves a series of steps and rules for constructing a tree-like structure to predict disease outcomes.

3.5 Random Forest

1. The motivation to use Random Forest algorithm
Random forests are an ensemble learning method that combines multiple decision trees to make predictions. They are widely used in various domains, including machine learning, data mining, and pattern recognition.
The main goals of the random forest algorithm are:
1. Improved predictive accuracy: Random forests aim to provide better prediction accuracy than individual decision trees by combining the predictions of multiple trees.
2. Reduced variance and increased stability: Random forests introduce randomness through bootstrap sampling, reducing the variance and making the predictions more stable.
3. Mitigating overfitting: Random forests help mitigate overfitting by aggregating the predictions of multiple trees, reducing the impact of individual noisy or outlier-prone trees.
4. Feature importance estimation: Random forests provide a measure of feature importance, helping identify the most relevant features for prediction.
5. Handling high-dimensional data: Random forests efficiently handle high-dimensional datasets by randomly selecting a subset of features at each split point.
6. Robustness to outliers and missing data: Random forests are robust to outliers and missing data due to the averaging process and the ability to handle missing values.
7. Efficiency and scalability: Random forests can be trained efficiently and scaled to handle large datasets, thanks to parallelization and computational optimizations.

2. Talk breifly about Random Forest algorithm
The random forest algorithm is an ensemble learning method that combines multiple decision trees to make predictions. It is a popular and powerful machine learning algorithm known for its versatility and robustness. Here's a brief overview of how the random forest algorithm works:
1. Data sampling: Random forests use bootstrap sampling, where multiple subsets of the original training data are created by randomly selecting data points with replacement. Each subset is used to train a decision tree.
2. Tree construction: For each subset of data, a decision tree is built using a technique called recursive partitioning. The tree is grown by recursively splitting the data based on selected features and split criteria, such as Gini impurity or information gain.
3. Random feature selection: At each split point of a decision tree, only a random subset of features is considered for splitting. This random feature selection helps reduce the correlation between trees and prevents dominance of a single feature.
4. Ensemble prediction: Once all the decision trees are constructed, predictions are made by aggregating the individual predictions of each tree. In classification tasks, the majority vote of the trees determines the final prediction. In regression tasks, the predictions are typically averaged across the trees.
5. Feature importance: Random forests provide a measure of feature importance by evaluating the impact of each feature on prediction accuracy. This is done by assessing the average decrease in impurity resulting from splitting on a particular feature across all the trees.

3. Formula of Random Forest
Gini Index use to select a feature to split further we need to know how impure or pure that split will be. A pure sub-split means that either you should be getting "yes" or "no". Suppose this is our dataset.

Gini Index $= 1 - \sum_{i=1}^{n} P_i^2 == 1 - [(P_+^2 + (P_-^2)]$

Where P_+ is the probability of a positive class and P_- is the probability of a negative class.

Entropy which is also used to measure the impurity of the split. The mathematical formula for entropy is:

$E(S) = -p_{(+)} \log p_{(+)} - p_{(-)} \log p_{(-)}$

Evaluation Matrix

The evaluation metrics provided in the code you shared are crucial for assessing the performance of model, particularly in the context of disease prediction. The reason we use these columns is that they provide a comprehensive view of the model's performance, allowing us to make more informed decisions. By looking at the TP, FP, TN, and FN values, we can calculate various evaluation metrics that give us a deeper understanding of the model's strengths and weaknesses.
Accuracy : measures the overall correctness of the predictions and is calculated as the ratio of correct predictions to the total number of predictions.

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

Precision : measures the accuracy of positive predictions. Precision indicates the proportion of correctly classified positive instances (diseased) out of all instances predicted as positive.

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

Recall (Sensitivity): measures the ability of the model to identify positive instances correctly. Recall represents the proportion of correctly classified positive instances (diseased) out of all actual positive instances.

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

Specificity((True Negative Rate)) : measures the ability of the model to identify negative instances correctly. Specificity represents the proportion of correctly classified negative instances (non-diseased) out of all actual negative instances.

$$Specificity = \frac{TN}{(TN + FP)}$$

F1 Score : is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall.

$$F1 \quad Score = 2 * \frac{(P * R)}{(P + R)}$$

3.6 Naive Bayes

1. The motivation to use Naive Bayes algorithm
The Naive Bayes algorithm is favored for its simplicity, speed, and effectiveness, particularly in text classification and spam detection. It is computationally efficient, making it suitable for large datasets and real-time applications. Despite its assumption of feature independence, which often doesn't hold, it performs well in practice. The algorithm is robust with small training datasets and provides interpretable probabilistic results. Its ability to handle both binary and multiclass classification problems further enhances its versatility. These characteristics make Naive Bayes a practical and powerful choice for various machine learning tasks.

2. Briefly about the Naive Bayes algorithm
The Naive Bayes algorithm is favored for its simplicity, speed, and effectiveness, particularly in text classification and spam detection. It is computationally efficient, making it suitable for large datasets and real-time applications. Despite its assumption of feature independence, which often doesn't hold, it performs well in practice. The algorithm is robust with small training datasets and provides interpretable probabilistic results. Its ability to handle both binary and multiclass classification problems further enhances its versatility. These characteristics make Naive Bayes a practical and powerful choice for various machine learning tasks.
3. Formula of Naive Bayes

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)} \tag{1}$$

$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c) \tag{2}$$

Above,

- $P(c \mid x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x \mid c)$ is the likelihood which is the probability of the *predictor* given *class*.
- $P(x)$ is the prior probability of the *predictor*.

3.7 K-Nearest Neighbors

1. The motivation to use K-Nearest Neighbors algorithm

The K-Nearest Neighbors (KNN) algorithm is motivated by its simplicity, intuitive concept, and effectiveness in various machine learning tasks. One primary motivation is its non-parametric nature, which means it doesn't make any assumptions about the underlying data distribution. This flexibility makes KNN suitable for both linear and non-linear relationships between features and labels. Another motivation is its ease of implementation and understanding. KNN requires minimal training time since it memorizes the entire training dataset, making it computationally efficient during inference. Moreover, its straightforward concept of classifying or predicting a new instance based on the majority vote or averaging of its nearest neighbors' labels makes it easily interpretable, making it particularly valuable in exploratory data analysis and model debugging. Furthermore, KNN performs well in scenarios where the decision boundary is irregular or when the data is noisy. Its ability to adapt to local structures in the data makes it robust to outliers and irrelevant features. Additionally, KNN can handle multi-class classification tasks seamlessly without the need for additional modifications. Despite its simplicity, KNN has limitations such as computational inefficiency with large datasets and the need to tune the hyperparameter k. Nevertheless, its strengths in handling diverse data distributions and ease of implementation make KNN a valuable addition to the machine learning toolbox, particularly in scenarios where interpretability and flexibility are prioritized.

2. Briefly about the K-Nearest Neighbors algorithm

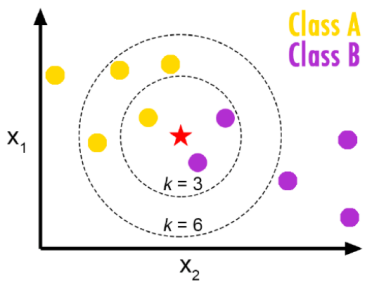
The K-Nearest Neighbors (KNN) algorithm is a simple, non-parametric, and lazy learning method used for classification and regression tasks. In KNN, the output for a given data point is determined by the majority class (for classification) or the average value (for regression) of its k nearest neighbors in the feature space. The algorithm operates based on distance metrics, typically Euclidean distance, to find the closest neighbors. KNN is easy to implement and understand, and it adapts naturally to multi-class classification problems. However, it can be computationally intensive for large datasets, as it requires calculating distances to all training samples. Despite this, its simplicity and effectiveness make it a popular choice for many practical applications.

3. Formula of K-Nearest Neighbors

Assume we are given a dataset where X is a matrix of features from an observation and Y is a class label. We will use this notation throughout this article. *k-nearest neighbors* then, is a method of classification that estimates the conditional distribution of Y given X and classifies an observation to the class with the highest probability. Given a positive integer k, k-nearest neighbors looks at the k observations closest to a test observation x_0 and estimates the conditional probability that it belongs to class j using the formula

$$Pr(Y = j \mid X = x_0) = \frac{1}{k} \sum_{i \in N_0} I(y_i = j)$$

where, $\mathcal{N}_0$ is the set of k-nearest observations and $I(y_i = j)$ is an indicator variable that evaluates to 1 if a given observation (x_i, y_i) in $\mathcal{N}_0$ is a member of class j, and 0 if otherwise. After estimating these probabilities, k-nearest neighbors assigns the observation x_0 to the class which the previous probability is the greatest. The following plot can be used to illustrate how the algorithm works:



- If we choose $K = 3$, then we have 2 observations in Class B and one observation in Class A. So, we classify the red star to Class B.
- If we choose $K = 6$, then we have 2 observations in Class B but four observations in Class A. So, we classify the red star to Class A.

Since in k-NN algorithm, we need k nearest points, thus, the first step is calculating the distance between the input data point and other points in our training data. Suppose x is a point with coordinates $(x_1, x_2, \ldots, x_p)$ and y is a point with coordinates $(y_1, y_2, \ldots, y_p)$, then the distance between these two points is:

$$d(x, y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

3.8 Support Vector Machine

1. The motivation to use Support Vector Machine Neighbors algorithm
Support Vector Machines (SVM) are motivated by their powerful ability to handle high-dimensional data and their effectiveness in binary and multiclass classification tasks. SVM aims to find the optimal hyperplane that maximizes the margin between different classes, which enhances the model's generalization capability. This robustness makes SVM particularly effective for complex datasets where clear class boundaries are needed. Additionally, SVM can employ various kernel functions to handle non-linear relationships, making it versatile across different types of data. Its capacity to manage overfitting through regularization parameters further strengthens its performance. These attributes—effective handling of high-dimensional spaces, versatility with kernels, and robustness against overfitting—make SVM a compelling choice for various machine learning challenges.

2. Briefly about the Support Vector Machine algorithm
Support Vector Machines (SVM) are supervised learning models used for classification and regression tasks. SVM works by finding the optimal hyperplane that separates data points of different classes with the maximum margin, which is the distance between the hyperplane and the nearest data points from each class (support vectors). This maximized margin helps improve the model's generalization ability. SVM can handle linear and non-linear classification problems by using kernel functions (e.g., polynomial, radial basis function) to transform the data into higher dimensions where a linear separator may be found. SVM is effective in high-dimensional spaces and versatile with different kernel functions, making it suitable for complex datasets with distinct class boundaries.

3. Formula of Support Vector Machine

Dual

$$\text{maximize } f(c_1, \ldots, c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i c_i (\mathbf{x}_i^\top \mathbf{x}_j) y_j c_j,$$

$$\text{subject to } \sum_{i=1}^{n} c_i y_i = 0, \text{ and } 0 \leq c_i \leq \frac{1}{2n\lambda} \text{ for all } i.$$

Sub-gradient descent

$$f(\mathbf{w}, b) = \left[\frac{1}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b)\right) \right] + \lambda \|\mathbf{w}\|^2.$$

4 Results and Discussions

4.1 Exploratory Data Analysis

Table 6: Descriptive Statistics of Symptoms in the Dataset

Symptom	Count	Mean	Std Dev	Min	50%	Max
Itching	4962	0.138049	0.344986	0.0	0.0	1.0
Skin Rash	4962	0.160016	0.366658	0.0	0.0	1.0
Nodal Skin Eruptions	4962	0.021967	0.146590	0.0	0.0	1.0
Continuous Sneezing	4962	0.045143	0.207639	0.0	0.0	1.0
Shivering	4962	0.021967	0.146590	0.0	0.0	1.0

Table 6 continued from previous page

Symptom	Count	Mean	Std Dev	Min	50%	Max
Chills	4962	0.162233	0.368702	0.0	0.0	1.0
Joint Pain	4962	0.139057	0.346041	0.0	0.0	1.0
Stomach Pain	4962	0.045143	0.207639	0.0	0.0	1.0
Acidity	4962	0.045143	0.207639	0.0	0.0	1.0
Ulcers on Tongue	4962	0.021967	0.146590	0.0	0.0	1.0
Muscle Wasting	4962	0.021967	0.146590	0.0	0.0	1.0
Vomiting	4962	0.389158	0.487608	0.0	0.0	1.0
Burning Micturition	4962	0.043934	0.204969	0.0	0.0	1.0
Spotting Urination	4962	0.021967	0.146590	0.0	0.0	1.0
Fatigue	4962	0.392785	0.488419	0.0	0.0	1.0
Weight Gain	4962	0.023176	0.150478	0.0	0.0	1.0
Anxiety	4962	0.023176	0.150478	0.0	0.0	1.0
Cold Hands and Feets	4962	0.023176	0.150478	0.0	0.0	1.0
Mood Swings	4962	0.046352	0.210268	0.0	0.0	1.0
Weight Loss	4962	0.092705	0.290047	0.0	0.0	1.0
Restlessness	4962	0.046352	0.210268	0.0	0.0	1.0
Lethargy	4962	0.092705	0.290047	0.0	0.0	1.0
Patches in Throat	4962	0.021967	0.146590	0.0	0.0	1.0
Irregular Sugar Level	4962	0.023176	0.150478	0.0	0.0	1.0
Cough	4962	0.114672	0.318657	0.0	0.0	1.0
High Fever	4962	0.276904	0.447514	0.0	0.0	1.0
Sunken Eyes	4962	0.021967	0.146590	0.0	0.0	1.0
Breathlessness	4962	0.091495	0.288341	0.0	0.0	1.0
Sweating	4962	0.137848	0.344775	0.0	0.0	1.0
Dehydration	4962	0.021967	0.146590	0.0	0.0	1.0
Indigestion	4962	0.045143	0.207639	0.0	0.0	1.0
Headache	4962	0.230552	0.421229	0.0	0.0	1.0
Yellowish Skin	4962	0.185409	0.388668	0.0	0.0	1.0
Dark Urine	4962	0.115881	0.320114	0.0	0.0	1.0
Nausea	4962	0.232971	0.422766	0.0	0.0	1.0
Loss of Appetite	4962	0.234180	0.423528	0.0	0.0	1.0
Pain Behind the Eyes	4962	0.024385	0.154258	0.0	0.0	1.0
Back Pain	4962	0.046352	0.210268	0.0	0.0	1.0
Constipation	4962	0.046352	0.210268	0.0	0.0	1.0
Abdominal Pain	4962	0.209794	0.407203	0.0	0.0	1.0
Diarrhoea	4962	0.114672	0.318657	0.0	0.0	1.0
Mild Fever	4962	0.071947	0.258426	0.0	0.0	1.0
Yellow Urine	4962	0.023176	0.150478	0.0	0.0	1.0
Yellowing of Eyes	4962	0.165861	0.371993	0.0	0.0	1.0
Acute Liver Failure	4962	0.023176	0.150478	0.0	0.0	1.0
Fluid Overload	4962	0.0	0.0	0.0	0.0	0.0
Swelling of Stomach	4962	0.023176	0.150478	0.0	0.0	1.0
Swelled Lymph Nodes	4962	0.070738	0.256412	0.0	0.0	1.0
Malaise	4962	0.142684	0.349786	0.0	0.0	1.0
Blurred and Distorted Vision	4962	0.069528	0.254376	0.0	0.0	1.0
Phlegm	4962	0.071947	0.258426	0.0	0.0	1.0
Throat Irritation	4962	0.024385	0.154258	0.0	0.0	1.0
Redness of Eyes	4962	0.024385	0.154258	0.0	0.0	1.0
Sinus Pressure	4962	0.024385	0.154258	0.0	0.0	1.0
Runny Nose	4962	0.024385	0.154258	0.0	0.0	1.0
Congestion	4962	0.024385	0.154258	0.0	0.0	1.0
Chest Pain	4962	0.141475	0.348546	0.0	0.0	1.0
Weakness in Limbs	4962	0.021967	0.146590	0.0	0.0	1.0
Fast Heart Rate	4962	0.047561	0.212858	0.0	0.0	1.0
Pain During Bowel Movements	4962	0.023176	0.150478	0.0	0.0	1.0
Pain in Anal Region	4962	0.023176	0.150478	0.0	0.0	1.0
Bloody Stool	4962	0.023176	0.150478	0.0	0.0	1.0
Irritation in Anus	4962	0.023176	0.150478	0.0	0.0	1.0
Neck Pain	4962	0.046352	0.210268	0.0	0.0	1.0
Dizziness	4962	0.068319	0.252318	0.0	0.0	1.0

Table 6 continued from previous page

Symptom	Count	Mean	Std Dev	Min	50%	Max
Cramps	4962	0.023176	0.150478	0.0	0.0	1.0
Bruising	4962	0.023176	0.150478	0.0	0.0	1.0
Obesity	4962	0.046352	0.210268	0.0	0.0	1.0
Swollen Legs	4962	0.023176	0.150478	0.0	0.0	1.0
Swollen Blood Vessels	4962	0.021967	0.146590	0.0	0.0	1.0
Puffy Face and Eyes	4962	0.023176	0.150478	0.0	0.0	1.0
Enlarged Thyroid	4962	0.024385	0.154258	0.0	0.0	1.0
Brittle Nails	4962	0.024385	0.154258	0.0	0.0	1.0
Swollen Extremeties	4962	0.024385	0.154258	0.0	0.0	1.0
Excessive Hunger	4962	0.093914	0.291738	0.0	0.0	1.0
Extra Marital Contacts	4962	0.021967	0.146590	0.0	0.0	1.0
Drying and Tingling Lips	4962	0.023176	0.150478	0.0	0.0	1.0
Slurred Speech	4962	0.024385	0.154258	0.0	0.0	1.0
Knee Pain	4962	0.023176	0.150478	0.0	0.0	1.0
Hip Joint Pain	4962	0.023176	0.150478	0.0	0.0	1.0
Muscle Weakness	4962	0.047561	0.212858	0.0	0.0	1.0
Stiff Neck	4962	0.046352	0.210268	0.0	0.0	1.0
Swelling Joints	4962	0.046352	0.210268	0.0	0.0	1.0
Movement Stiffness	4962	0.023176	0.150478	0.0	0.0	1.0
Spinning Movements	4962	0.021967	0.146590	0.0	0.0	1.0
Loss of Balance	4962	0.069528	0.254376	0.0	0.0	1.0
Unsteadiness	4962	0.023176	0.150478	0.0	0.0	1.0
Weakness of One Body Side	4962	0.021967	0.146590	0.0	0.0	1.0
Loss of Smell	4962	0.024385	0.154258	0.0	0.0	1.0
Bladder Discomfort	4962	0.023176	0.150478	0.0	0.0	1.0
Foul Smell of Urine	4962	0.020758	0.142587	0.0	0.0	1.0
Continuous Feel of Urine	4962	0.023176	0.150478	0.0	0.0	1.0
Passage of Gases	4962	0.023176	0.150478	0.0	0.0	1.0
Internal Itching	4962	0.023176	0.150478	0.0	0.0	1.0
Toxic Look (Typhos)	4962	0.023176	0.150478	0.0	0.0	1.0
Depression	4962	0.047561	0.212858	0.0	0.0	1.0
Irritability	4962	0.096332	0.295076	0.0	0.0	1.0
Muscle Pain	4962	0.096332	0.295076	0.0	0.0	1.0
Altered Sensorium	4962	0.023176	0.150478	0.0	0.0	1.0
Red Spots Over Body	4962	0.047763	0.213286	0.0	0.0	1.0
Belly Pain	4962	0.023176	0.150478	0.0	0.0	1.0
Abnormal Menstruation	4962	0.048771	0.215410	0.0	0.0	1.0
Dischromic Patches	4962	0.021967	0.146590	0.0	0.0	1.0
Watering From Eyes	4962	0.021967	0.146590	0.0	0.0	1.0
Increased Appetite	4962	0.024385	0.154258	0.0	0.0	1.0
Polyuria	4962	0.024385	0.154258	0.0	0.0	1.0
Family History	4962	0.046554	0.210702	0.0	0.0	1.0
Mucoid Sputum	4962	0.023176	0.150478	0.0	0.0	1.0
Rusty Sputum	4962	0.024385	0.154258	0.0	0.0	1.0
Lack of Concentration	4962	0.023176	0.150478	0.0	0.0	1.0
Visual Disturbances	4962	0.023176	0.150478	0.0	0.0	1.0
Receiving Blood Transfusion	4962	0.024385	0.154258	0.0	0.0	1.0
Receiving Unsterile Injections	4962	0.024385	0.154258	0.0	0.0	1.0
Coma	4962	0.024385	0.154258	0.0	0.0	1.0
Stomach Bleeding	4962	0.024385	0.154258	0.0	0.0	1.0
Distention of Abdomen	4962	0.023176	0.150478	0.0	0.0	1.0
History of Alcohol Consumption	4962	0.023176	0.150478	0.0	0.0	1.0
Fluid Overload.1	4962	0.023176	0.150478	0.0	0.0	1.0
Blood in Sputum	4962	0.024385	0.154258	0.0	0.0	1.0
Prominent Veins on Calf	4962	0.023176	0.150478	0.0	0.0	1.0
Palpitations	4962	0.024385	0.154258	0.0	0.0	1.0
Painful Walking	4962	0.046352	0.210268	0.0	0.0	1.0
Pus Filled Pimples	4962	0.021967	0.146590	0.0	0.0	1.0
Blackheads	4962	0.021967	0.146590	0.0	0.0	1.0
Scurring	4962	0.021967	0.146590	0.0	0.0	1.0

Table 6 continued from previous page

Symptom	Count	Mean	Std Dev	Min	50%	Max
Skin Peeling	4962	0.023378	0.151115	0.0	0.0	1.0
Silver Like Dusting	4962	0.023176	0.150478	0.0	0.0	1.0
Small Dents in Nails	4962	0.023176	0.150478	0.0	0.0	1.0
Inflammatory Nails	4962	0.023176	0.150478	0.0	0.0	1.0
Blister	4962	0.023176	0.150478	0.0	0.0	1.0
Red Sore Around Nose	4962	0.023378	0.151115	0.0	0.0	1.0
Yellow Crust Ooze	4962	0.023176	0.150478	0.0	0.0	1.0

The dataset comprises various symptoms experienced by patients, with each symptom quantified across several statistical measures. Here is a summary of the key statistics calculated for each symptom:

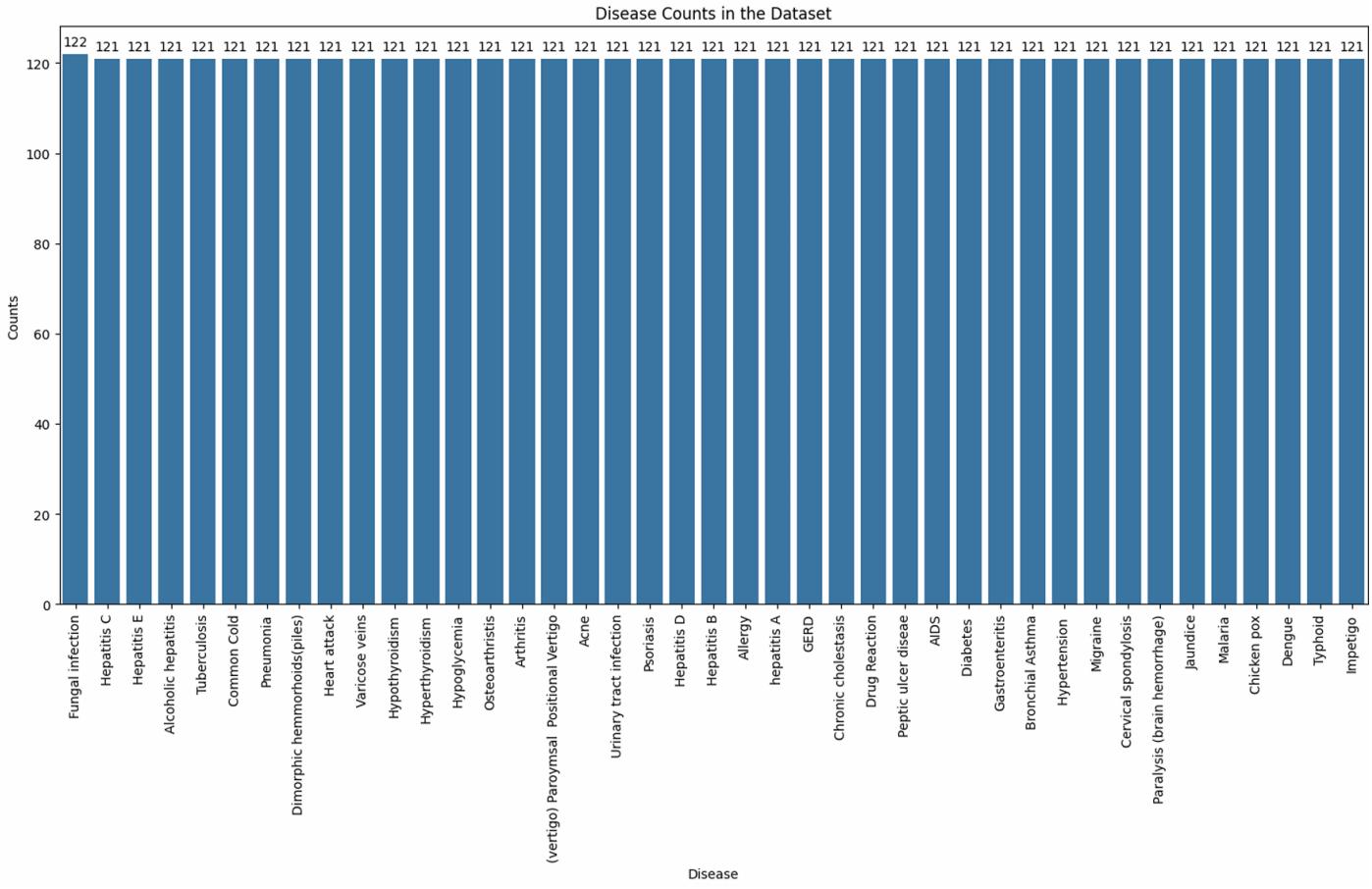
- **Count:** Each symptom has a total of 4962 observations, suggesting a complete dataset with no missing values.
- **Mean:** This is the average presence of the symptom across the dataset, indicating the proportion of the dataset that exhibits this symptom.
- **Standard Deviation (Std Dev):** Indicates the variability or spread of the symptom's presence across the dataset. A higher standard deviation signifies that the symptom occurrences are more spread out over the dataset.
- **Min:** The minimum value for each symptom is 0, which indicates that there are cases where these symptoms are completely absent.
- **25%, 50% (Median), and 75%:** These quartiles demonstrate that for most symptoms, 75% of the dataset or more do not exhibit the symptom, as these quartiles are typically 0.
- **Max:** The maximum value is 1 for all symptoms, showing that the symptoms are binary-encoded (0 or 1), where 1 represents the presence of a symptom.

Symptoms with Notable Mean and Standard Deviation

Symptom	Mean	Standard Deviation
Vomiting	0.389	0.488
Fatigue	0.393	0.488
High Fever	0.277	0.448
Loss of Appetite	0.234	0.424
Nausea	0.233	0.423
Yellowish Skin	0.185	0.389
Headache	0.231	0.421
Dark Urine	0.116	0.320
Cough	0.115	0.319
Diarrhoea	0.115	0.319

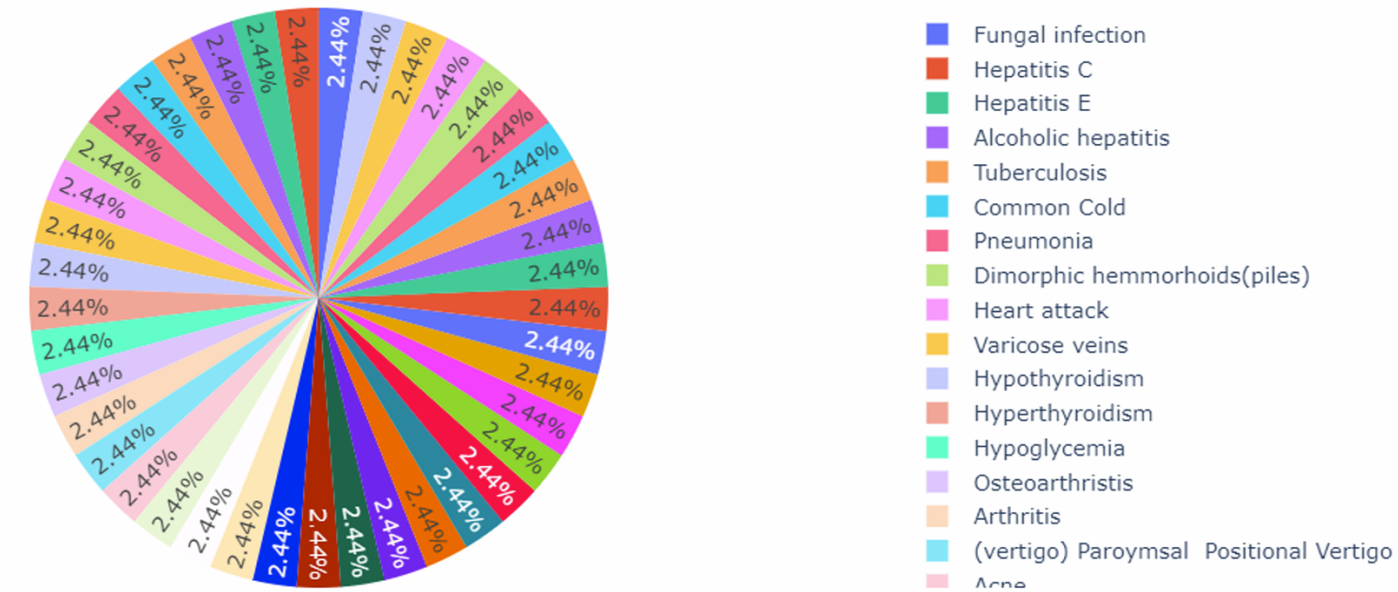
These statistics are crucial for understanding the distribution and frequency of each symptom in the dataset, aiding in further analysis such as correlation study or predictive modeling.

Count the occurrences of each unique value in the prognosis column

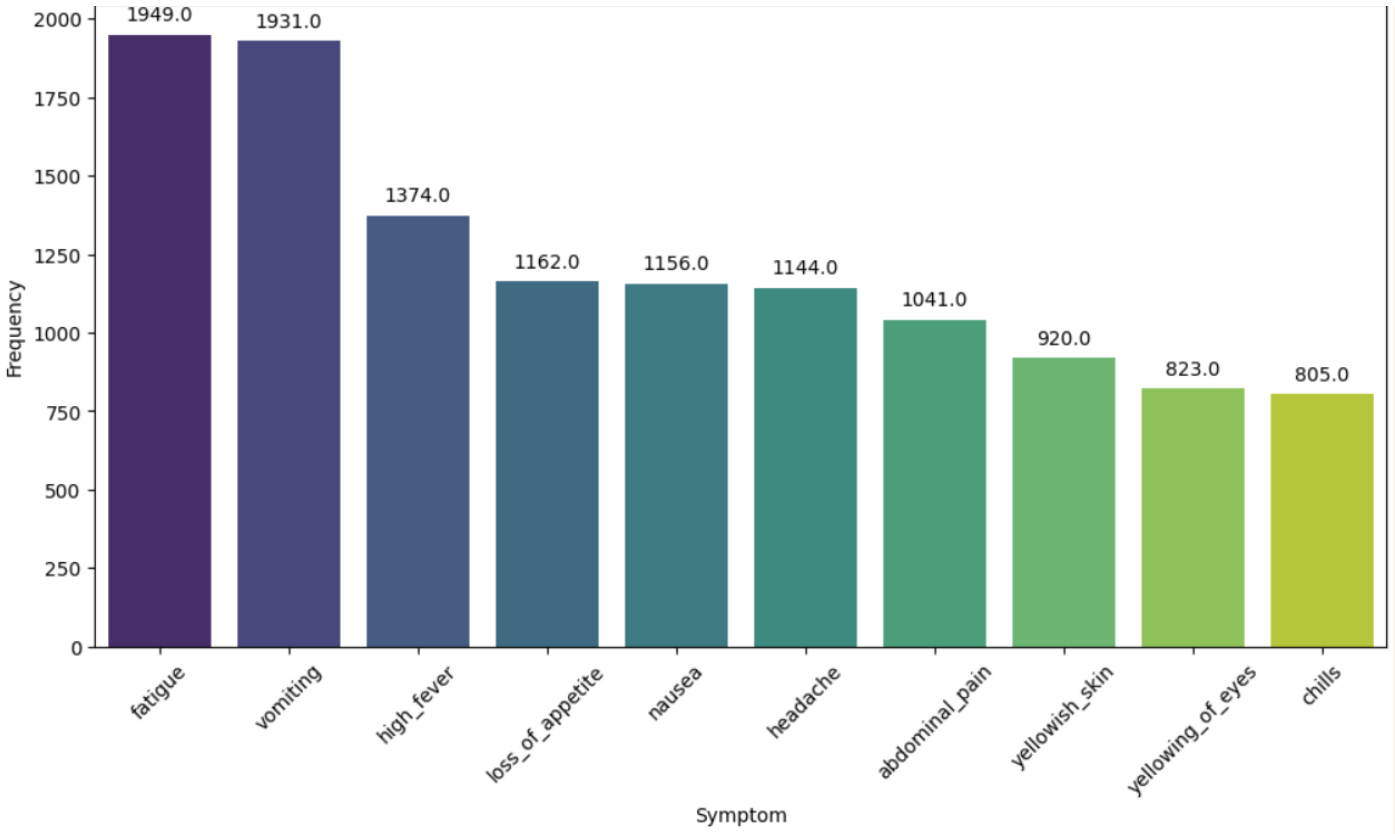


The bar chart above visualizes the distribution of different diseases within a medical dataset. Each bar represents the number of occurrences for a specific disease, with all conditions having a similar frequency. This uniform distribution suggests that the dataset is well-balanced across various diseases, which is beneficial for analytical consistency and training predictive models.

Prognosis Distribution

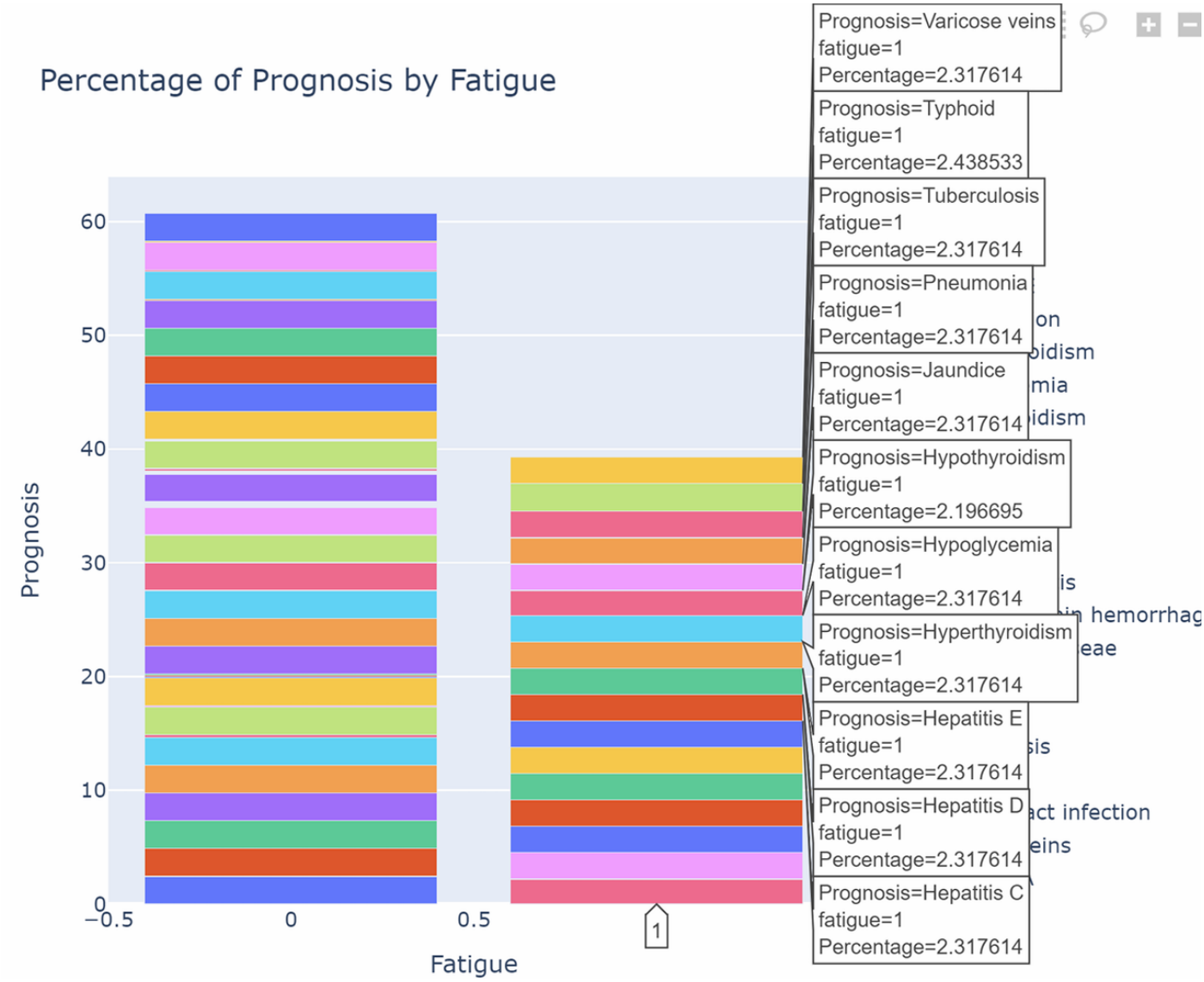


The pie chart above illustrates the proportion of cases for various diseases within a medical dataset. Each segment of the pie chart represents a different disease, labeled with both the disease name and its proportion of the total cases. The visualization shows a balanced distribution among the diseases, each constituting approximately 2.44% of the dataset. This equal distribution across different conditions suggests that the dataset is ideally structured for facilitating comparative analyses and building balanced predictive models.



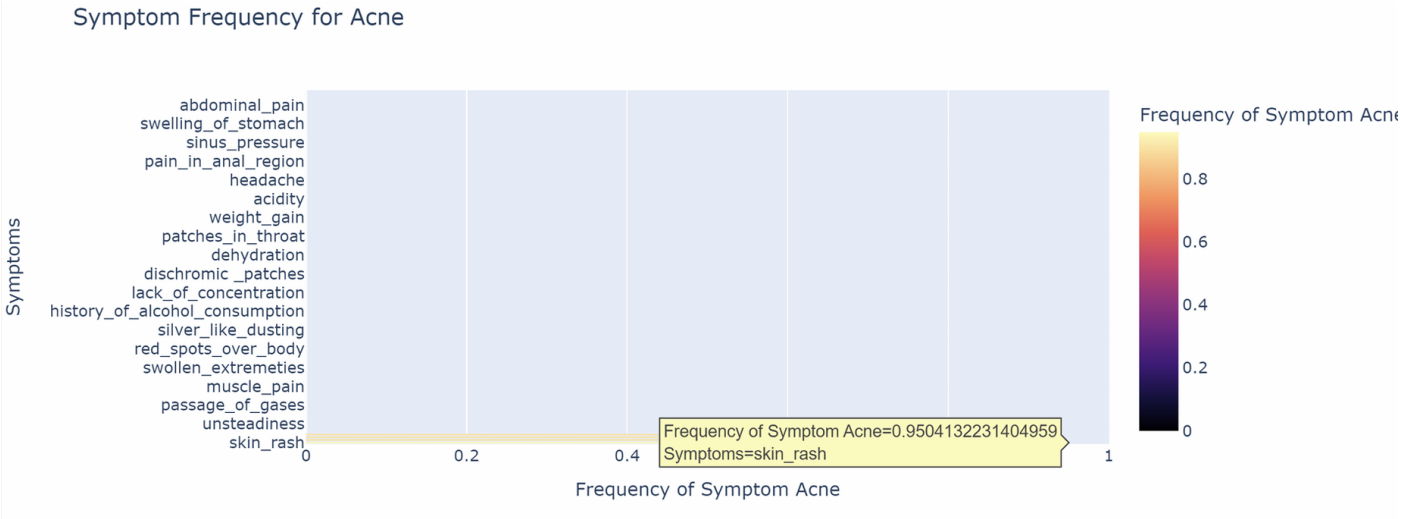
The bar chart above presents the frequency of the top 10 symptoms recorded in the medical dataset. The symptoms 'Fatigue' and 'Vomiting' appear most frequently, with over 1900 cases each, indicating their common occurrence in the dataset's conditions. The chart provides a clear visual indication of the symptoms' prevalence, emphasizing the importance of these symptoms in clinical diagnostics and treatment decisions.

Percentage of Prognosis by Fatigue



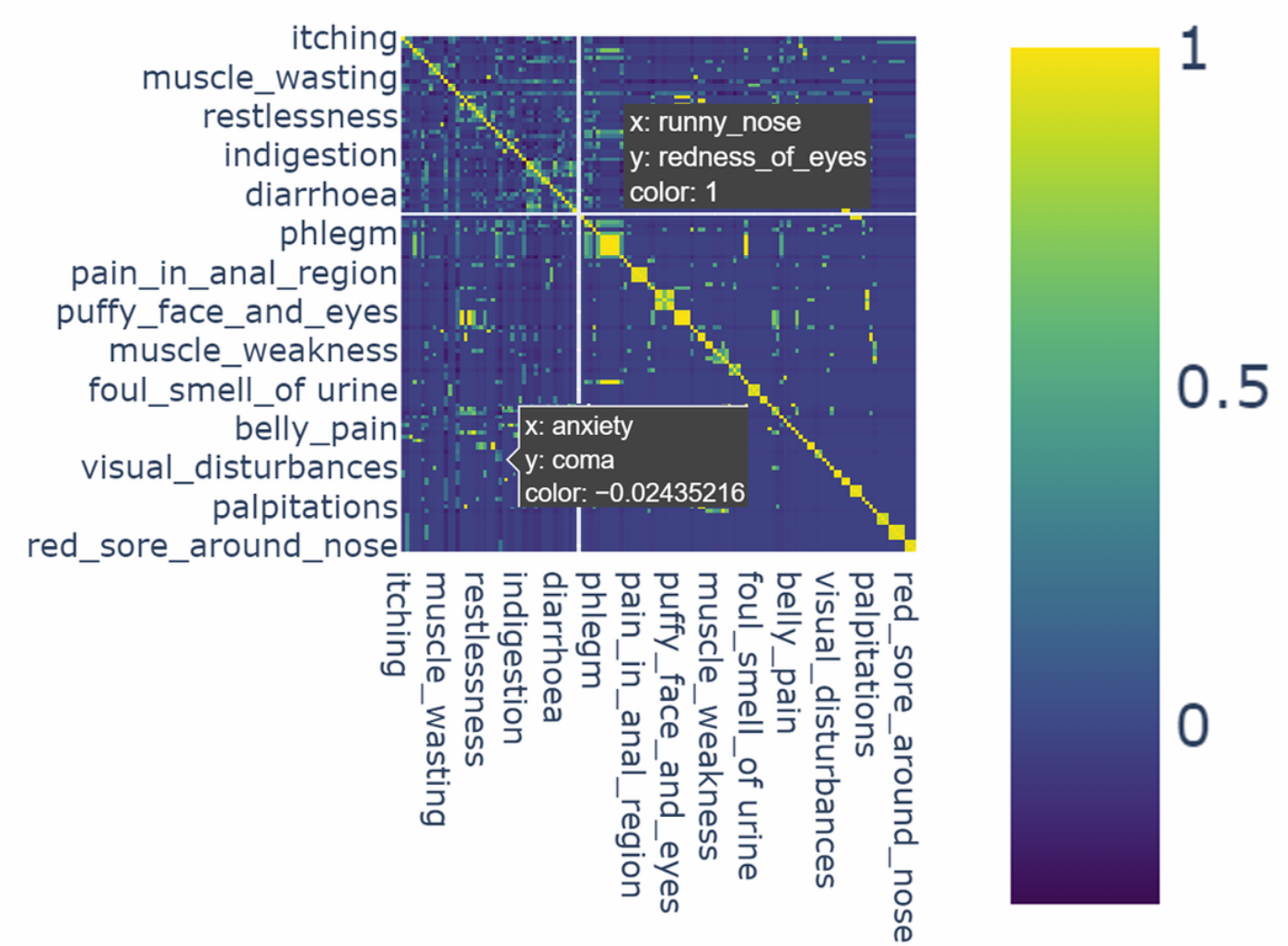
The stacked bar chart illustrates the percentage of various prognoses associated with the symptom of fatigue within a medical dataset. Each segment in the bars represents a different condition, colored distinctly for easy differentiation. The chart provides insight into how prevalent fatigue is across various diseases, highlighting its significance as a common symptom in multiple conditions. This visualization helps in understanding the broad impact of fatigue on patient diagnosis and management in clinical settings.

Symptom Frequency for Acne



This heatmap illustrates the frequency of various symptoms associated with acne within a medical dataset. Each row represents a different symptom, such as 'skin rash', 'muscle pain', and 'headache', among others. The color intensity varies from light to dark, indicating the frequency from low to high, respectively. 'Skin rash' shows the highest correlation with acne, highlighted by the darkest color. This visualization aids in identifying which symptoms are most commonly associated with acne, providing valuable insights for clinical assessments and diagnostic processes.

Correlation Matrix of Symptoms



The heatmap above displays the correlation matrix for various symptoms recorded in a medical dataset. Each cell represents the correlation coefficient between two symptoms, ranging from -1 to 1. Positive values (yellow) indicate a

direct correlation, while negative values (blue) suggest an inverse relationship. The heatmap is crucial for understanding which symptoms frequently co-occur, aiding in pattern recognition and the potential clustering of related symptoms for diagnostic analysis.

Symptom Correlation Analysis

Runny Nose and Redness of Eyes

Correlation Coefficient: 1
The symptoms *runny nose* and *redness of eyes* have a correlation coefficient of 1, indicating a perfect positive correlation. This means that when a runny nose is reported in the dataset, redness of the eyes is also consistently reported. This strong correlation can be attributed to their common association with conditions like allergies or the common cold, where both symptoms are typically present as part of the body's response to allergens or infections.

Anxiety and Coma

Correlation Coefficient: -0.02435216
On the other hand, the symptoms *anxiety* and *coma* show a slight negative correlation of approximately -0.02435216. Although close to zero, this negative value suggests a very weak inverse relationship between these symptoms. Practically, this indicates that occurrences of anxiety and coma do not tend to happen together frequently. Anxiety is usually a high-alert, conscious state, often associated with stress or nervous system hyperactivity, whereas coma represents a severe, unresponsive state of unconsciousness, often resulting from significant neurological damage or dysfunction.

Implications

Understanding these correlations is crucial for clinical diagnostics and research. For example, the strong link between *runny nose* and *redness of eyes* could be used to predict one symptom from the presence of the other, potentially speeding up preliminary diagnoses in clinical settings. Conversely, the lack of correlation between *anxiety* and *coma* underscores the distinct and unrelated pathways these symptoms generally represent in patient presentations, which might be important for differential diagnoses or when reviewing patient symptoms in emergency settings.

4.2 Model Evaluation Results

4.3 Results

	Accuracy	Precision	Recall	Specificity	F1 Score
Logistic Regression	1.0	1.0	1.0	1.0	1.0
Random Forest	1.0	1.0	1.0	1.0	1.0
Decision Tree	1.0	1.0	1.0	1.0	1.0
Naive Bayes	1.0	1.0	1.0	1.0	1.0
Support Vector Classifier	1.0	1.0	1.0	1.0	1.0
K-NeighborsClassifier	1.0	1.0	1.0	1.0	1.0

4.4 Discussion

The table above presents the performance metrics for various machine learning algorithms including Logistic Regression, Random Forest, Decision Tree, Naive Bayes, Support Vector Classifier, and K-Neighbors Classifier. The metrics evaluated are Accuracy, Precision, Recall, Specificity, and F1 Score. All algorithms achieved a perfect score of 1.0 across all metrics.

4.5 Nature of the Algorithms

- **Logistic Regression**: A linear model for binary classification that estimates probabilities using the logistic function.
- **Random Forest**: An ensemble learning method that constructs multiple decision trees and outputs the mode of their predictions.
- **Decision Tree**: A model that splits data into branches to make predictions based on decision rules.
- **Naive Bayes**: A probabilistic classifier based on Bayes' theorem with strong (naive) independence assumptions between the features.

- **Support Vector Classifier**: A classifier that finds the hyperplane which best separates the data into classes.
- **K-Neighbors Classifier**: A non-parametric method that classifies based on the majority class among the k-nearest neighbors.

4.6 Discussion of Results

The perfect scores obtained by all the models suggest that the dataset might be very well-structured and possibly less complex, making it easy for these algorithms to achieve high performance. While it is unusual for multiple algorithms to achieve perfect scores across all metrics, it is plausible under certain conditions:

- The dataset may have well-separated classes with clear decision boundaries, facilitating high accuracy for all models.
- The dataset might be balanced with an equal distribution of classes, reducing the risk of class imbalance affecting performance.
- The features in the dataset could be highly informative and relevant, providing strong signals for classification.

4.7 Comparison with Previous Studies

In previous studies, it is often observed that different algorithms have varying strengths and weaknesses depending on the dataset. For instance, Random Forest and Decision Trees generally perform well with structured data due to their ability to handle non-linear relationships. Support Vector Machines are effective with high-dimensional spaces, while Naive Bayes is known for its efficiency with large datasets and text classification.
The results obtained here may align with those seen in less complex, well-prepared datasets where feature selection and preprocessing have been optimized. However, it is essential to validate these results with cross-validation and test on different datasets to ensure the models are not overfitting.

5 Conclusion

The performance metrics indicate excellent model performance on the given dataset. Future work should include testing these models on additional datasets and employing cross-validation to ensure generalizability and robustness of the results.

6 Conclusions and Recommendations

The disease prediction using machine learning underscores the remark able potential of this interdisciplinary approach in revolutionizing healthcare. By harnessing vast datasets and advanced algorithms, machine learning techniques have demonstrated significant strides in accurately predicting various diseases, ranging from cardiovascular con ditions to infectious outbreaks. The synthesis of diverse methodologies, including feature selection, ensemble learning, and deep learning architectures, has enabled researchers to achieve high prediction accuracies and clinical relevance. However, challenges such as data heterogeneity, model interpretability, and ethical considerations remain pertinent areas for further investigation. Moving forward, concerted efforts in refining algorithms, integrating multi-modal data sources, and fostering collaboration between data scientists and healthcare professionals will be essential in realizing the full transformative impact of machine learning in disease prediction, ultimately enhancing early diagnosis, treatment efficacy, and patient outcomes. I chose the random forest method in my projects for several reasons:

1. **Accuracy and Robustness:**
    - Random forests tend to provide high accuracy due to their ability to build multiple decision trees and aggregate their results. This ensemble approach reduces the risk of overfitting compared to using a single decision tree.
2. **Handling Overfitting:**
    - By averaging the predictions of multiple trees, random forests can mitigate the overfitting issue that is often seen with individual decision trees. This makes random forests more generalizable to unseen data.
3. **Versatility:**
    - Random forests can be used for both classification and regression tasks, making them versatile for various types of projects.
4. **Feature Importance:**
    - One of the strengths of random forests is their ability to estimate the importance of each feature in the prediction. This helps in understanding the underlying data and can be useful for feature selection.
5. **Robustness to Noise:**

- Random forests are robust to noisy data and can handle a large number of input features without a significant risk of overfitting.

6. **Non-Parametric Nature:**
   - Being a non-parametric method, random forests do not assume any underlying distribution for the data, making them flexible and effective in many real-world scenarios.
7. **Efficiency with Large Datasets:**
   - They can handle large datasets efficiently, especially when implemented with parallel processing, as each tree can be built independently of the others.
8. **Ease of Use:**
   - The algorithm is relatively straightforward to use and implement with many machine learning libraries providing optimized versions of random forests.

7 Limitations and Future Works

7.1 Limitations

- **Dataset Limitations:** The study's reliance on a homogeneous Kaggle dataset may not adequately reflect the global diversity of health conditions, potentially limiting the broader applicability of findings.
- **Model Simplification:** Potential oversimplification in modeling may not capture complex interactions within health data, leading to less accurate predictions.
- **Feature Selection:** Current feature selection methods might overlook critical predictors due to reliance on predefined data sets, impacting the accuracy of disease predictions.
- **Validation Needs:** Models primarily validated on retrospective data may not perform consistently in prospective or real-world settings.

7.2 Future Works

- **Data Diversification:** Integrating diverse, multi-source datasets will be crucial to enhance the models' generalizability and robustness.
- **Advanced Modeling:** Exploring more sophisticated machine learning techniques, such as deep learning and advanced ensemble methods, could address current limitations in handling complex data relationships.
- **Dynamic Feature Engineering:** Implementing advanced feature engineering techniques could help in discovering vital new predictors that enhance model performance.
- **Real-world Testing:** Conducting validation in real-world clinical settings is essential to ascertain the models' practical utility and effectiveness.

- Support Vector Machines **?**. - K-Nearest Neighbors (KNN) **?**. - Naive Bayes Explained **?**. - A related article1 **?**. - A related article2 **?**. - A related article3 **?**. - A related article4 **?**. - A related article5 **?**.

References

[1]K. Gaurav et al. "Human Disease Prediction using Machine Learning". In: *International Journal of Industrial Engineering* (2023). URL: https://www.ije.ir/article_169090_5525e34b7bd485c6f9f9cc710f62522f.pdf.

[2]C. K. Gomathy and A. Rohirt Naidu. "The Prediction of Disease Using Machine Learning". In: *International Journal of Scientific Research in Engineering and Management* (2021). URL: https://www.researchgate.net/publication/357449131_THE_PREDICTION_OF_DISEASE_USING_MACHINE_LEARNING.

[3]Sneha Grampurohit and Chetan Sagarnal. "Disease Prediction using Machine Learning Algorithms". In: *Industry, Healthcare and Bioscience* (2020). URL: https://ieeexplore.ieee.org/abstract/document/9154130.

[4]Neha Gupta, Kriti Gandhi, and Shafali Dhall. "Disease Prediction using Machine Learning". In: *International Journal for Research in Applied Science and Engineering Technology* (2020), p. 502. URL: https://www.researchgate.net/publication/343883157_Disease_prediction_using_machine_learning.

[5]LibreTexts contributors. *K-Nearest Neighbors (KNN)*. https://stats.libretexts.org/Bookshelves/Computing_and_Modeling/RTG:_Classification_Methods/3:_K-Nearest_Neighbors_(KNN). 2023.

[6]V. Sharon Rose. "Disease Prediction Using Machine Learning". In: *Bachelor of Technology* (2021). URL: https://www.karunya.edu/aqar/2020-21/QIF/Criteria_1/1.3.4/520.pdf.

[7]Analytics Vidhya. *Naive Bayes Explained*. https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/. 2017.

[8]Wikipedia contributors. *Support Vector Machine*. https://en.wikipedia.org/wiki/Support_vector_machine. 2023.