Predicting Diseases from Symptoms Using Machine Learning

CIND820: Capstone Project

Project by: Divya Arora
divya.arora@torontomu.ca
Student #500334710

Supervisor: Tamer Abdou

Ryerson University

Table of Contents

Abstract

With the current world situation, environment and lifestyle affects an individual's health greatly and makes them prone to various diseases. In the rat race to provide for themselves and their families, individuals experience stress at both home and workplace (Kanakaraddi et al., 2021). This has an adverse effect on their health, leading to majority of untimely diseases. However, advancement in technology enables early detection of diseases such as arthritis, asthma. The early detection and prevention of diseases is of utmost importance with the ongoing stressful lives that people are experiencing and can help prevent many life threatening diseases (Kanakaraddi et al., 2021).

The main aim of this project is to solve health-related issues by supporting doctors to more accurately predict and diagnose diseases using machine learning techniques, based on symptoms that patients may be experiencing, making timely treatment a possibility which benefits patient care (Grampurohit & Sagarnal, 2020). Can the implementation of a supervised machine-learning model be used to determine if diseases can be accurately predicted and diagnosed?

Using the proposed Disease Prediction dataset that contains 132 parameters, we will conduct a series of modeling on those predictor variables to determine the class variable, prognosis. This research will explore the important features selected and apply them to test against several machine learning algorithms to compare their performance.

The project will emphasize on the technique of Classification and Regression and knowledge discovery. Classification model will be used to predict the type of disease (as a dependent column) from the type of symptoms (as independent columns). For predictive modeling, the proposed algorithms used are (1) Logistic Regression, (2) Random Forest, (3) Naïve Bayes, (4) Support Vector Classifier and (5) K-nearest neighbor. These algorithms will be implemented using Python programming language.

Keywords: Early detection, Training and Testing dataset, Classification and Regression, Data Mining and Knowledge Discovery

Research Questions

- Can we accurately predict disease using machine learning?
- Which feature selection technique will be appropriate for this dataset?
- Which machine learning algorithms are most effective in producing reliable results?
- What method of cross validation evaluation will be employed?
- Can the ML algorithms' settings be tuned for the optimal performance?

Introduction

The use of machine learning algorithms for disease detection and prediction has made significant advancements in recent years in the healthcare industry. The creation of machine learning algorithms that can accurately predict diseases based on the symptoms displayed by patients is of great interest. The traditional method of disease diagnosis relies on the knowledge of medical experts who, in order to make informed decisions evaluate patient symptoms, medical history, and diagnostic tests (Mahata et al., 2023). Factors like human error, time consumption, cost and limited availability restrict the accuracy of human diagnostics (Mahata et al., 2023). On the other hand, machine learning algorithms can process huge amounts of data and spot complex patterns improving diagnostic accuracy and can assist medical experts in making more informed decisions, leading to early detection and personalized treatment plans (Mahata et al., 2023).

The idea behind machine learning-based disease prediction models is making use of historical patient data. The dataset in this project consists of patient's symptoms and the corresponding disease diagnosis. When this data is fed into a machine learning algorithm, the model can learn to recognize patterns and associations between symptoms and diseases.

Therefore, as advancements in technology and data availability continue, machine learning models for disease prediction using symptoms hold great potential to transform healthcare by enhancing diagnostic accuracy, improving patient outcomes and improving global healthcare (Mahata et al., 2023). Moreover, it can assist the medical experts to prioritize urgent cases and ultimately reduce mortality rates.

The data set was retrieved from:
(1) https://impact.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html (The New York Presbyterian Hospital provides a database of health symptoms that individuals experienced along with the corresponding prognosis admitted during 2004. (*Disease*, n.d.))
(2) https://github.com/anujdutt9/Disease-Prediction-from-Symptoms
(3) https://www.kaggle.com/datasets/neelima98/disease-prediction-using-machine-learning

The raw data and processes for this study can be accessed from:
https://github.com/DA-CIND/CIND820

Literature Review

This section describes previously published articles on predictive analysis as well as the methods the authors used in predicting diseases. The reviewed papers investigate algorithms such as Logistic Regression, Random Forest, Naïve Bayes, Support Vector Classifier and K-nearest neighbor and use accuracy as a performance metric which is relevant to the dataset being utilised in this project.

As per the article, “Multiple disease prediction using Machine learning algorithms”, in order to assess the efficacy of therapeutic medicines, data mining for healthcare is an interdisciplinary study field that emerged from database statistics. Diabetes-related heart disease is one form of heart disease that affects diabetics (Arumugam et al., 2021). Diabetes is a chronic disease that arises from either inadequate insulin production by the pancreas or inappropriate insulin utilisation by the body. Cardiovascular disease, commonly known as heart disease, is a group of illnesses that harm the heart or blood arteries (Arumugam et al., 2021). There are several data mining classification methods for predicting heart disease, however not enough data exists to predict heart disease in a person with diabetes. Since the decision tree model consistently outperformed the naive Bayes and support vector machine models, the authors optimised it for the best performance in identifying the risk of heart disease in people with diabetes (Arumugam et al., 2021).

The article, “Stratification of Parkinson Disease using python scikit-learn ML library”, talks about whether a patient has parkinsons disease or is in good health (Kolte et al., 2019). Parkinson's disease is a condition that affects the central nervous system and affects how well the body moves. The symptoms of this chronic illness get worse with time and are commonly experienced by older people. This illness may be examined using general machine learning techniques, which offer varying degrees of accuracy (Kolte et al., 2019). The Parkinson disease dataset with repeated acoustic characteristics had 48 associated features for 240 people. Whether the patient has Parkinson's disease or is in good health is indicated by the target "status" column, which has a value of 0 or 1. Since healthcare applications typically require more precision and cannot be compromised, the ideal application is chosen with the maximum degree of accuracy possible. The naive Bayes classifier, gradient boosting, and support vector machines are the main models applied in this research (Kolte et al., 2019). By evaluating the characteristics of the patients, these methods can be quite effective for doctors in predicting the condition. The dataset was subjected to the chosen techniques and principal component analysis, and the following degrees of accuracy were attained: Decision tree (71.3%), Support Vector Machine (81.2%), and Logistic Regression (78.7%). Gaussian Naive Bayes classifier has the highest accuracy (86.25%) out of all the tested models (Kolte et al., 2019).

The article, “Disease Prediction using Machine Learning Algorithms” is similar to this project as the researcher’s dataset is similar to my dataset but their work is not being completely replicated as the code for this project is written in python and their source code is not available in their article. In the later section of this report, proposed work and methodology, algorithms are listed that are proposed to be used such as Logistic Regression, Random Forest, KNN, Naïve Bayes and Support Vector Classifier. Whereas, machine learning algorithms used by (Grampurohit & Sagarnal, 2020b) are Decision Tree, Random Forest, Naïve Bayes.

Medical database analysis that is accurate helps with early disease identification, patient care, and social support. Machine learning techniques have been successfully used in many fields, including the diagnosis and prognosis of diseases. By assisting clinicians in early disease prediction and diagnosis, a classifier system developed using machine learning algorithms seeks to considerably aid in the resolution of health-related issues (Grampurohit & Sagarnal, 2020b). The authors created a disease prediction system utilising machine learning techniques such as

Decision Tree classifier, Random forest classifier, and Naive Bayes (Grampurohit & Sagarnal, 2020b).

The medical records of 4920 patients who were at risk for 41 diseases due to a cluster of symptoms were used to train the algorithm. 95 out of 132 symptoms have been considered in order to avoid overfitting. Using the K fold cross validation methodology (K=5), the authors evaluated the efficacy of each strategy against the dataset. They came to the conclusion that all three approaches perform incredibly well on the dataset based on their findings. However, Nave Bayes may be slightly superior to the other two algorithms in terms of performance (Grampurohit & Sagarnal, 2020b). It is evident from the historical development of machine learning and its uses in the medical field that methodologies and techniques have developed that make it possible to analyse complex data in an easy and basic manner. The effectiveness of three algorithms on a medical record is carefully compared in this study. Results from each algorithm were up to 95% accurate (Grampurohit & Sagarnal, 2020b).

The article, "Symptoms Based Disease Prediction Using Decision Tree and Electronic Health Record Analysis" shows that in order to distinguish between the patient's illness and more common symptoms, health professionals conduct far too many disease surveys and collect information on patients, the severity of their illness, and their symptoms (Radhika et al., 2020). As a result, useful information hidden in the data set is used to train the model that predicts the disease based on the symptoms. The authors constructed the decision tree classifier model, which is trained using the dataset in a shorter amount of time, after normalising the data using case gradient descent standardisation techniques. After normalisation, their trained model is utilised to predict the disease, along with the degree of confidence, underlying causes, and suggested preventive measures (Radhika et al., 2020).

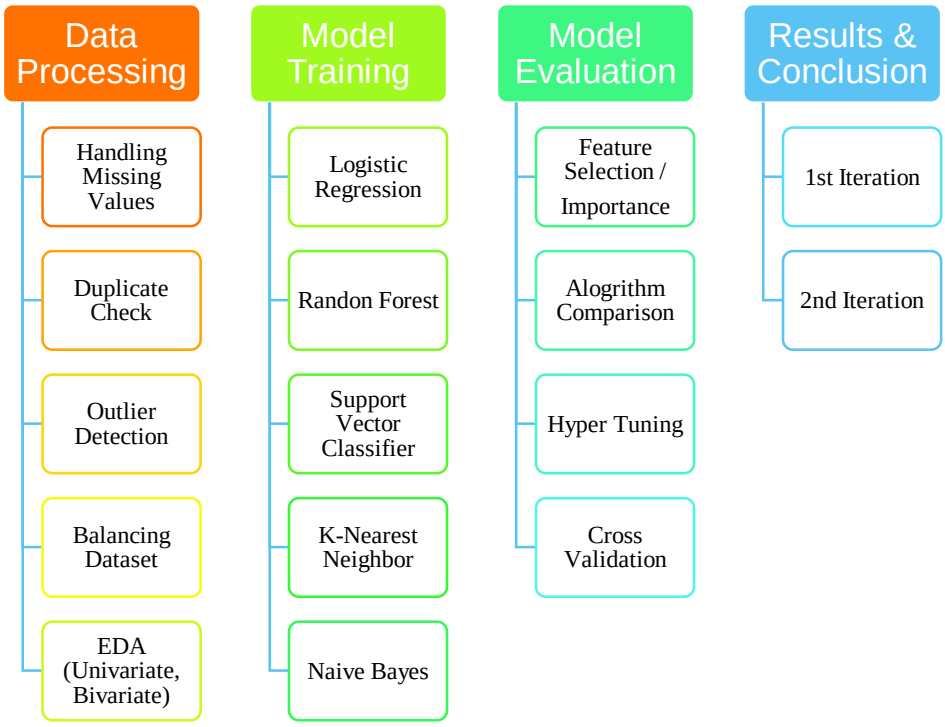
On the other hand, their technology takes a EHR (Electronic Health Records) as input and converts it into a text file (Radhika et al., 2020). The text file is summarised using the NLTK (Natural Language Toolkit) to help the patient understand the health report. They used health record analysis in this article, which provides customised input and user interaction with the system, to increase confidence (Radhika et al., 2020). To make this system a fully functional health monitoring system, extra modules are offered in addition to the two primary modules (Radhika et al., 2020). These modules include functions for locating doctors, planning doctor visits, and saving and retrieving medical records. The entire system consists of two modules, one for disease prediction and the other for health monitoring. The second module is utilised as one of the training data sets to get the best outcome and boost confidence. In order to collect the user's input and provide status updates, this procedure is linked to the user interface. Because the files users supply as input may contain native languages, multilingual summarization makes it straightforward to quickly summarise health records from many parts of the world (Radhika et al., 2020). Currently, only English is used to write the document. This article defines disease prediction and examines some of the associated tasks, such as appointment scheduling and locating the closest medical institution, in addition to defining disease prediction utilising highly personalised training data sets (Radhika et al., 2020).

Disorders affecting the heart, kidneys, breast and brain are the focus of machine learning models (KNN, NB, DT, CNN, SVM, and LR) to assess for disease detection in the article "Disease Prediction Using Machine Learning". The goal of their study is to investigate the idea that

supervised machine learning algorithms can improve healthcare by quickly and correctly identifying diseases (Ferjani, 2020). Accuracy was the most important performance metric, and the three most frequently used prediction algorithms in the literature were SVM, RF, and LR. The CNN model performed the best at predicting common diseases (Ferjani, 2020). Additionally, the SVM model constantly showed greater accuracy for kidney diseases and PD because of its dependability in handling high-dimensional, semi-structured, and unstructured data (Ferjani, 2020). RF showed advantage in the likelihood of correctly identifying the prediction of breast cancer because to its potential to scale efficiently for large datasets and susceptibility to prevent overfitting. To finish, the LR algorithm was the most accurate at predicting heart illnesses (Ferjani, 2020).

Approach

Figure 1: Approach Process



To begin with, initial preprocessing and cleaning of the raw data was performed. This process included data cleaning, handling missing values, checking duplicates, outlier detection and balancing dataset. In order to understand the relationship between variables (symptoms) and their impact on predicting the occurrence of a disease, techniques such as univariate and bivariate analysis were used as part of Exploratory Data Analysis (EDA). The goal of EDA is to uncover patterns, trends, or distributions within the dataset by visualizing the data through plots, charts, or summary statistics.

In order to determine the models' accuracy and reliability, it is essential to evaluate their performance. Therefore, MCC and Brier Score were used as standard metrics for binary classification.

Once the data was explored and preprocessed, modeling was conducted using SKlearn package.

Feature Selection aims to choose a subset of input variables by removing features that provide no predictive information. The feature selection method is split into three categories: 1) filters; 2) wrappers; and 3) embedded methods (Gupta, 2023).

For the purpose of this study the following Feature Selection techniques were used: 1) filter (chi-square, ANOVA); 2) wrapper (forward selection, backward elimination, recursive feature elimination); 3) embedded (decision tree based) (Gupta, 2023).

Next step was Feature Importance. Feature importance quantifies the importance of each feature compared to the other features in the model. It helps identify the most influential features relative to the others. It represents the contribution of each feature in the models decision making process (Feature Importance | Machine Learning in the Elastic Stack [8.8] | Elastic, n.d.).

Five algorithms were used for modeling: Logistic Regression, Random Forest, Support Vector Classifier, K-Nearest Neighbor and Gaussian Naïve Bayes algorithm. The modelling process was repeated twice, once with default parameters and the second after hyper-tunning the parameters.

After modelling, cross validation strategies such as Leave one out, Stratified K Fold, and Holdout Method using Logistic Regression and Support Vector are explored in this paper.

Data Description

The dataset used in this project has the following attributes:

- The size of the dataset is 133 columns and 4920 rows. The dataset has been bifurcated into training and testing data. The model will be trained through training dataset and the result will be tested on testing dataset. In this dataset, the 133 columns are split as follows:
    - 132 columns have continuous discrete binary values (Symptoms wherein 0 stands for no symptom and 1 stands for presence of symptom in the patient.)
    - 1 column is qualitative discrete categorical (Prognosis of disease consisting of 41 diseases such as Dengue, Diabetes, Heart attack etc.)

The rows consist of continuous variables i.e. dummy variable taking the value of 0 (false) and 1 (true). The diseases listed in this dataset range from communicable diseases (spread from one person to another) to non communicable diseases (do not spread to others).

Table 1: Attribute summary

Attribute	Type
Symptoms (all 133 columns)	Continuous (binary: 0, 1)
Prognosis	Categorical

Below is a list of all 41 diseases:

Table 2: List of Prognosis

Prognosis	Prognosis	Prognosis
Pneumonia	Acne	Dimorphichemorrhoids (piles)
Heart attack	Psoriasis	Varicose veins
Migraine	Drug Reaction	Hyperthyroidism
Paralysis (brain hemorrhage)	AIDS	Osteoarthritis
Malaria	Gastroenteritis	(vertigo) Paroymsal Positional Vertigo
Dengue	Hypertension	Urinary tract infection
Impetigo	Cervical spondylosis	Hepatitis D
Hepatitis B	Jaundice	Allergy
hepatitis A	Chicken pox	GERD
Chronic cholestasis	Typhoid	Bronchial Asthma
Peptic ulcer disease	Diabetes	Hepatitis C
Fungal infection	Hypothyroidism	Alcoholic hepatitis
Hepatitis E	Hypoglycemia	Common Cold
Tuberculosis	Arthritis	

Data Preprocessing

Data cleaning or preprocessing is the process of preparing and transforming the raw data before it is used for training a machine learning model. The goal is to ensure the data is in a suitable format, free from errors, and ready for analysis. The steps involved in data cleaning and preprocessing for a disease prediction model are:

Handling Missing Values: Missing values are a common issue in datasets and can affect the performance of the prediction model. In the training dataset, while analysing the data a column was discovered named as unnamed133, which had enough NaN values. Since NaN values were significantly high we omitted the entire column from the dataset. Other than this there were no missing values in the dataset.

Checking Duplicates: Identification of the duplicates can be done by comparing the values of each record across different columns or using unique identifiers, such as patient IDs or timestamps. As there was no unique identifier present in the dataset, an assumption has been made that all the records are unique. The total number of records in the training dataset are 4920 and after removing duplicates, the number of unique records were 304.

After running all the possible exploration techniques on the unique records (304) in the training dataset, including Feature Selection, Feature Importance, Matthew's Correlation Coefficient, Model Building (Logistic Regression, Random Forest, KNN, SVM, Naïve Bayes) and hyperparameter tuning, the accuracy turned out to be 100%. See below table. Therefore, there is no scope for improvement.

Table3: Model and its accuracy

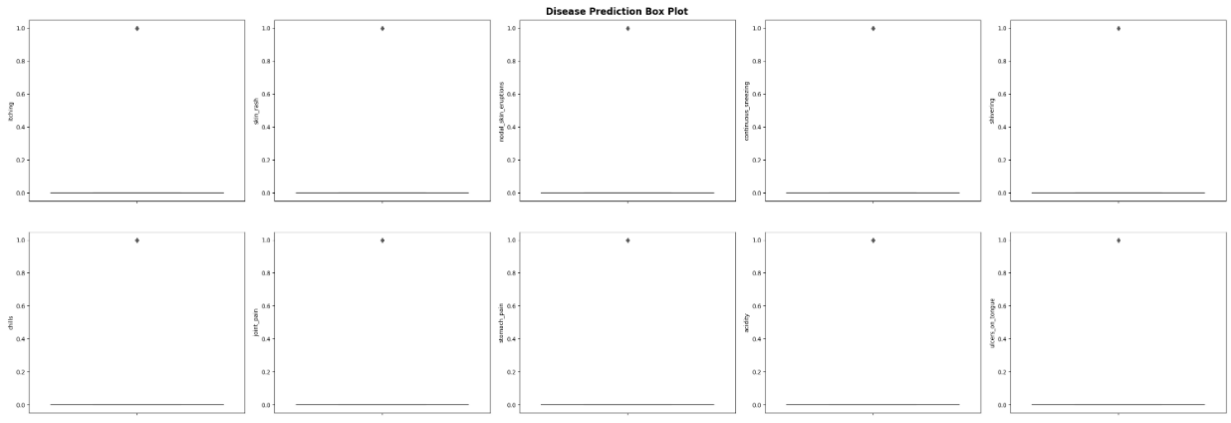
Model	Accuracy
Logistic Regression	1.0
Random Forest	1.0
K Nearest Neighbor	1.0
SVM	1.0
Gaussian Naïve Bayes	1.0

In the test dataset, all the records (42) are unique, no duplicates were found.

So, in this project all the records are considered including duplicate records with an assumption of them being unique. The code of this can be found at https://github.com/DA-CIND/CIND820/blob/main/Duplicate%20Check.ipynb.

Outlier Detection and Treatment: Outliers are extreme values that deviate significantly from the majority of the data. Outliers can impact the training of the prediction model and affect its performance. Using box-plot, we can see the data distribution of these variables and clearly identify that there are no outlier data points outside minimum and maximum whiskers within all variables.

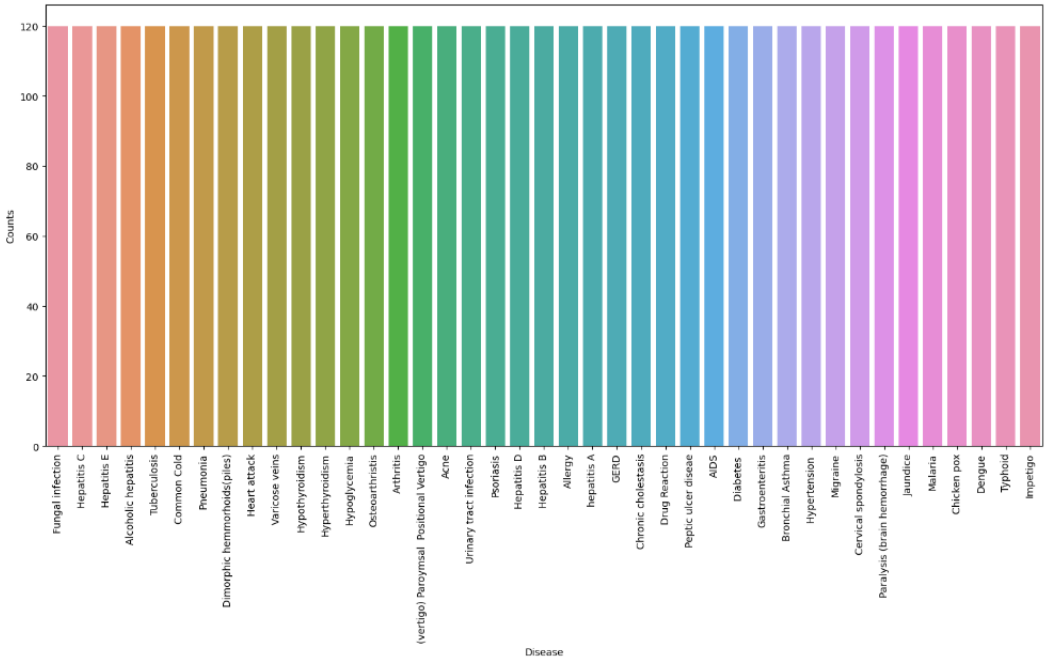
Figure 2: Box-plot of Outlier



Balancing the Dataset: Disease prediction models often rely on a subset of relevant features or variables. By analyzing the dataset, the most informative features that are likely to contribute to the prediction task can be identified. This helps in reducing dimensionality, and avoiding irrelevant or redundant variables that may introduce unnecessary complexity.

To check whether the dataset is balanced or not, using value_counts of dependent variable "prognosis", the below plot is formed, where it can be observed that the dataset is a balanced dataset i.e. there are exactly 120 samples for each disease, and no further balancing is required.

Graph 1: Balanced Dataset



Splitting the Dataset: While importing the dataset from the source, two separate files i.e. training_csv and testing_csv were downloaded. The assumption is that the given data was randomly partitioned into two independent sets giving an impression that the partition was performed in accordance with Holdout Method.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) plays a vital role in disease prediction using machine learning algorithms. EDA focuses on understanding the relationship between symptoms and prognosis, exploring the data patterns, and extracting meaningful insights to build accurate predictive models.

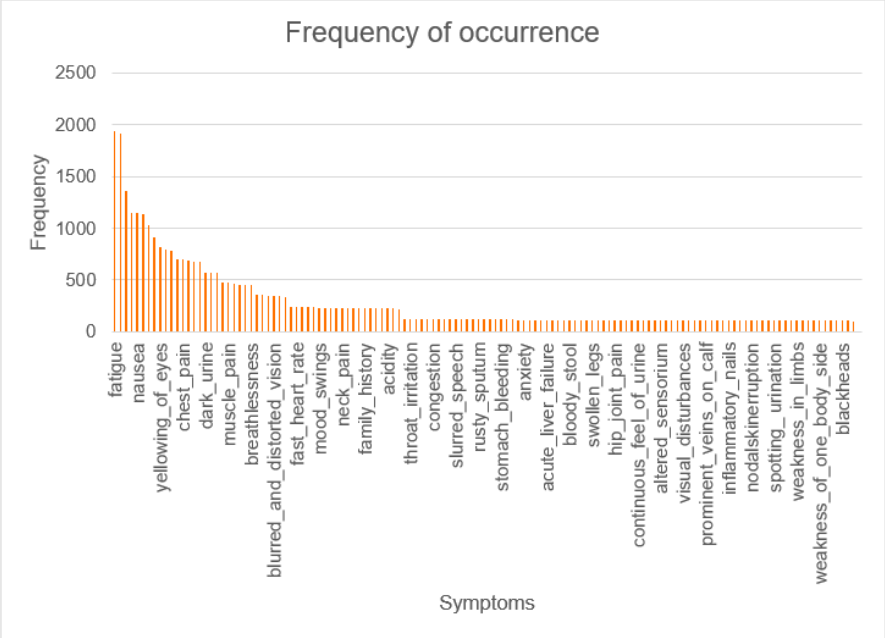
Data visualization techniques were used to explore the relationships between symptoms and prognosis. Correlations or patterns that may exist between symptoms and prognosis can be identified with the help of charts, plots, bar graph and subplots. These visualizations help in and generating insights that can drive feature selection and model development.

Initial Analysis

The frequency of occurrence of symptoms plays an important role in disease prediction models and might indicate the presence or severity of an illness. Some diseases may be more

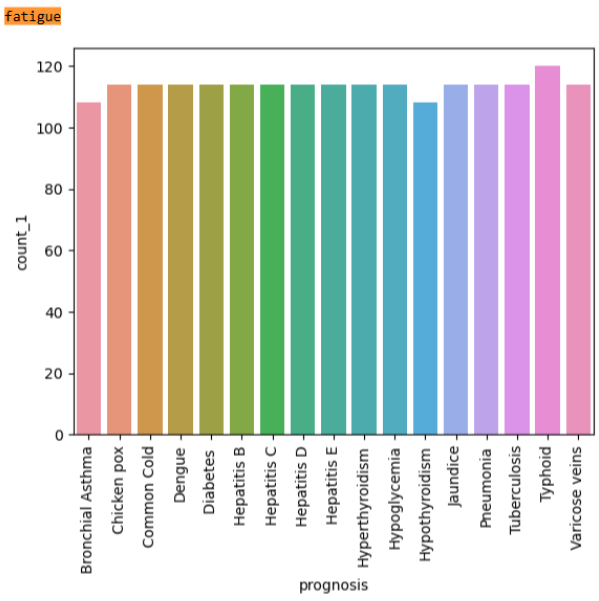
frequently linked to particular symptoms. If a particular symptom occurs frequently in a dataset, it can indicate a higher likelihood of that symptom being indicative of the disease of interest.
The graph below shows that fatigue is the most prevalent symptom in diseases followed by nausea, yellowing of eyes etc.

Graph 2: Frequency of Occurrence



The below barplot visualized through Seaborn library of Python shows a list of all the diseases that have fatigue as a common symptom such as chicken pox, common cold, diabetes, dengue, typhoid, and others.

Graph 3: Diseases that have Fatigue as a symptom



The symptom “fluid_overload” did not occur in any of the diseases and frequency of occurrence of the symptom “fatigue” was 1932 times in the dataset.

Figure 3: Value count of symptoms

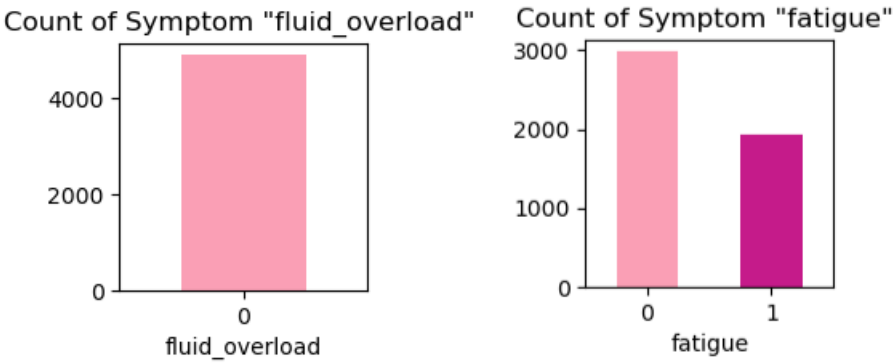
```
fluid_overload
==============================

0    4920
Name: fluid_overload, dtype: int64
```

```
fatigue
==============================

0    2988
1    1932
Name: fatigue, dtype: int64
```

Figure 4: Bar graph showing value count of symptoms

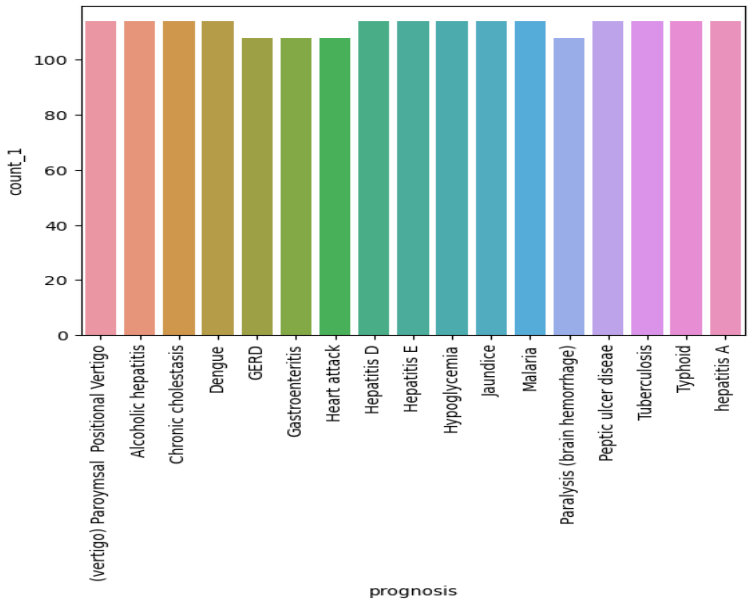


In the context of disease prediction datasets, univariate and bivariate analyses are statistical techniques used to understand the relationship between variables and their impact on predicting the occurrence of a disease.

Univariate Analysis:

Univariate analysis focuses on analyzing a single variable at a time. This analysis helps identify patterns, outliers, and understanding variable's characteristics. Graphical representations like histograms or box plots are used to summarize and visualize the data. For visualization, seaborn library and matplotlib are used to show barplot of symptom vs prognosis.

Graph 4: Vomiting (Symptom)



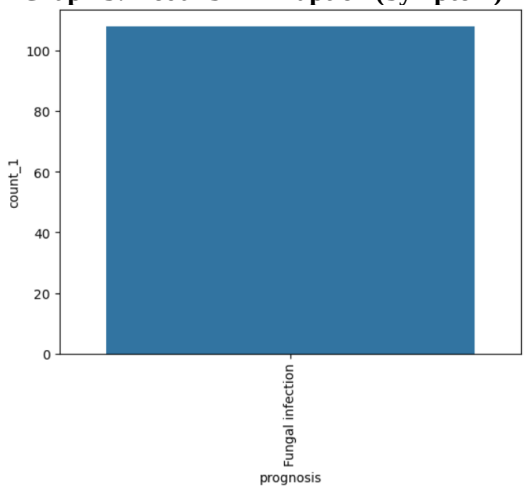
In graph 4, count of symptom is shown on the y axis whereas prognosis is shown on x axis. It shows that there are some symptoms which are common to a lot of disease. For example, Vomiting is a common symptom in: (vertigo) Paroymsal Positional Vertigo, Alcoholic Hepatitis, Chronic Cholestasis, Dengue, GERD, (Hepatitis A, D, E), Malaria, Tuberculosis etc.

Similarly, Itching is a common symptom in Chicken Pox, Chronic Cholestasis, Drug Reaction, Fungal Infection, Hepatitis B, Jaundice and Fatigue is a common symptom in: Chicken Pox, Common Cold, Diabetes, Dengue, Typhoid etc.

This shows that these symptoms are common in many diseases irrespective of the disease being communicable or non communicable. So, a pattern cannot be observed from this analysis.

Now moving on to the analysis of the symptoms which are unique in some diseases which are as follows:

Graph 5: Nodal Skin Eruption (Symptom)



In graph 5, it can be seen that nodal skin eruption is unique to Fungal infection. Similarly, Continuous_Sneezing is a symptom of allergy and common cold, Muscle_wasting is a symptom of AIDS, Irregular_sugar_level is a symptom of diabetes and Pain_behind_eyes are a symptom of dengue.

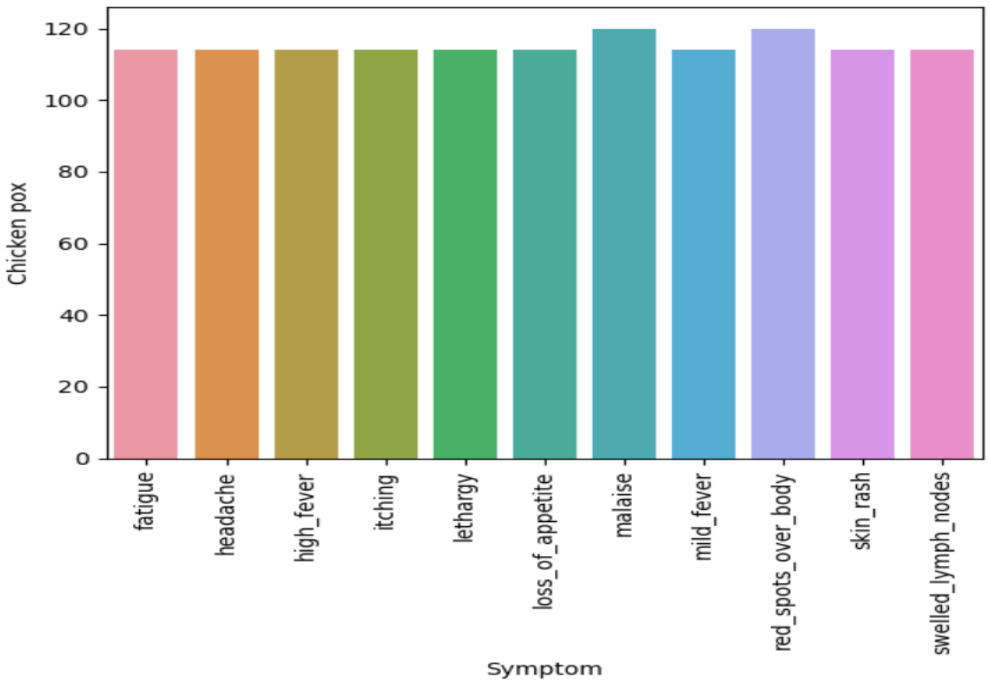
This shows that some symptoms are unique to a particular kind of disease as described above. Therefore, it can be concluded that from a unique symptom, early detection of disease can be predicted.

Bivariate Analysis:

Bivariate analysis involves examining the relationship between two variables in the dataset. It helps determine how one variable may influence or relate to another and is useful for identifying associations between a potential predictor variable and the disease outcome.

For the purpose of bivariate analysis, visualization techniques used are seaborn and Matplotlib. In graph 6, prognosis is represented on y axis and symptoms are represented on x axis.

Graph 6: Chicken Pox (Prognosis)



The above graph shows that a patient suffering from Chicken pox experiences symptoms such as fatigue, headache, high fever, itching and lethargy etc. Similarly, patients suffering from Arthiritis experience movement_stiffness, muscle_weakness, painful_walking, stiff neck, swelling_joints.

Correlation

Correlation is a statistical measure that indicates the extent to which two variables are related. A correlation coefficient of +1 indicates a perfect positive correlation, while a correlation

coefficient of -1 indicates a perfect negative correlation. A correlation coefficient of 0 indicates no correlation.

In disease prediction dataset, correlation can be used to identify features that are related to the disease. Features that are highly correlated with the disease are more likely to be useful for predicting the disease.

There are a number of ways to explore correlation. In this study, correlation matrix, Heatmap, Matthews Correlation and Brier Score (Comotto, 2022) were examined.

Correlation Matrix:

A correlation matrix is a table that shows the correlation coefficients between all pairs of variables in the dataset. This can be a useful way to get an overview of the relationships between the different variables. As there are 132 columns, only a section of the correlation matrix is shown in the below figure.

Figure 5: Correlation matrix

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity
itching	1.000000	0.318158	0.326439	-0.086906	-0.059893	-0.175905	-0.160650	0.202850	-0.086906
skin_rash	0.318158	1.000000	0.298143	-0.094786	-0.065324	-0.029324	0.171134	0.161784	-0.094786
nodal_skin_eruptions	0.326439	0.298143	1.000000	-0.032566	-0.022444	-0.065917	-0.060200	-0.032566	-0.032566
continuous_sneezing	-0.086906	-0.094786	-0.032566	1.000000	0.608981	0.446238	-0.087351	-0.047254	-0.047254
shivering	-0.059893	-0.065324	-0.022444	0.608981	1.000000	0.295332	-0.060200	-0.032566	-0.032566
...	...	...	...	...	...	...	...	...	...
small_dents_in_nails	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	0.359845	-0.033480	-0.033480
inflammatory_nails	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	0.359845	-0.033480	-0.033480
blister	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480
red_sore_around_nose	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480
yellow_crust_ooze	-0.061573	0.331087	-0.023073	-0.033480	-0.023073	-0.067765	-0.061889	-0.033480	-0.033480

132 rows × 132 columns

The correlation matrix in Figure 5 shows the correlation values between different symptoms. Looking at the matrix, each symptom is compared to every other symptom in terms of their correlation.

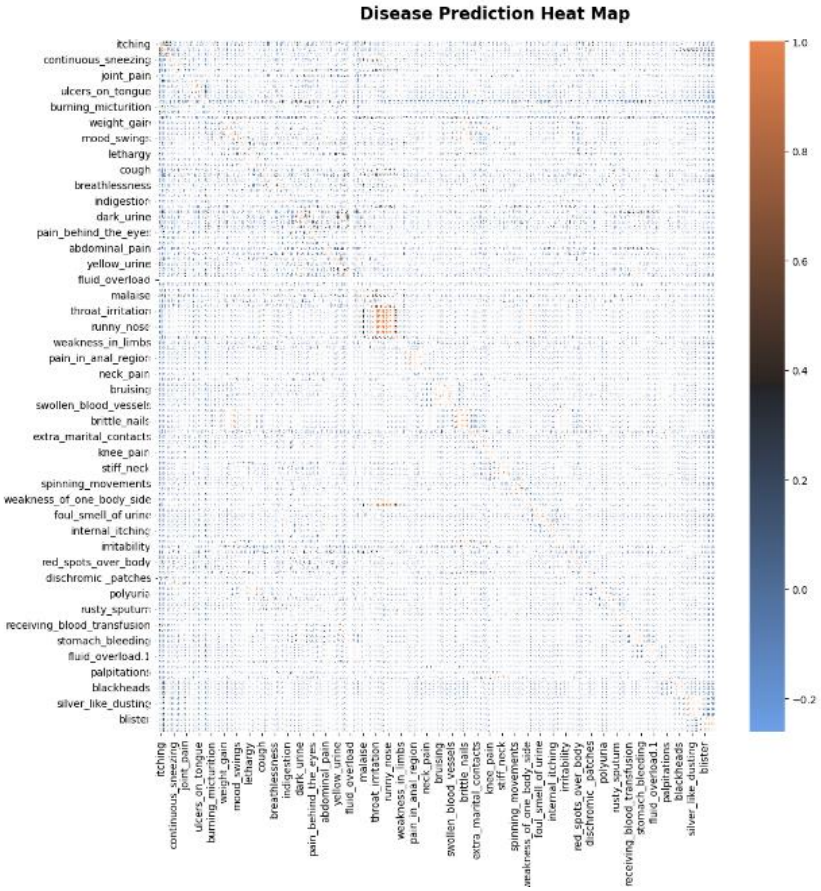
For example, the correlation between "itching" and "skin_rash" is 0.318158, indicating a positive correlation between these two symptoms. Similarly, the correlation between "itching" and "blister" is -0.061573, indicating a weaker negative correlation between these two symptoms.

By examining the correlation matrix, one can identify patterns and relationships between symptoms. Positive correlations suggest that the symptoms tend to occur together, while negative correlations suggest an inverse relationship, where the presence of one symptom is independent of the other.

Heat Map:

A heatmap was visualized to determine any correlation between any of the predictor variables and the target variable.

Figure 6: Heatmap displaying correlation values

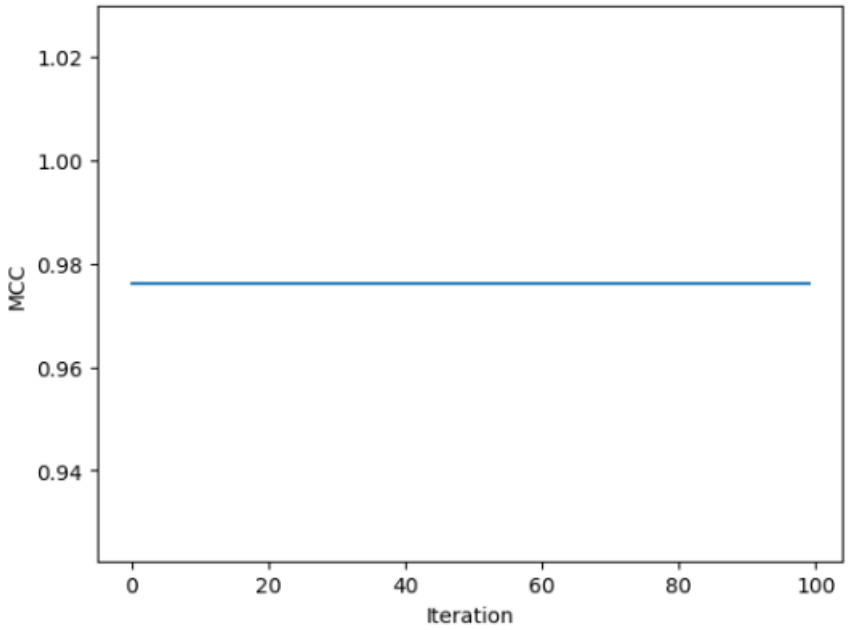


The faded orange diagonal line shows equality of correlation between variables which is not clearly visible so no meaningful inferences can be drawn. Unfortunately, there are too many attributes in the dataset so it is difficult to identify any strong relationships.

Matthews Correlation Coefficient:

The Matthews Correlation Coefficient (MCC) is a measure of the quality of a binary classification model. It takes into account true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) to assess the performance of the model. The MCC ranges from -1 to 1, with 1 indicating a perfect classification, 0 indicating a random classification, and -1 indicating a completely incorrect classification ("Matthews Correlation Coefficient," 2022).

Figure 7: Matthews Correlation Coefficient



In Figure 7, an MCC of 0.9762 is very close to 1, which suggests that the disease prediction model has achieved a high level of accuracy and is performing well. It indicates a strong positive correlation between the predicted and actual classifications. Overall, it signifies that the model's predictions align very closely with the ground truth and that it is an effective classifier for the disease outcome.

Brier Score:

The Brier score measures the accuracy of probabilistic predictions (Zach, 2020). In the context of this disease prediction model, it quantifies the quality of the symptom as a predictor for the prognosis (disease outcome). A lower Brier score indicates better predictive performance (Zach, 2020).

Looking at the results obtained, symptoms like "blister" and "red_sore_around_nose" have the lowest Brier scores (0.0024), suggesting that they are strong indicators of the disease outcome. On the other hand, symptoms like "fluid_overload," "foul_smell_of_urine," "nodal_skin_eruptions," "shivering," "ulcers_on_tongue," "muscle_wasting," and "spotting_urination" have slightly higher Brier scores (between 0.0231 and 0.0451), indicating lower predictive accuracy compared to the first two symptoms.

In summary, symptoms with lower Brier scores are more strongly correlated with the prognosis, indicating better predictive power, while symptoms with higher Brier scores may have lower accuracy.

To conclude, no meaningful inference could be drawn from the Heatmap as the number of columns are significantly high. Matthews Correlation and Brier Score as performance measures were more beneficial as compared to Heatmap.

Feature Selection

For the purpose of this study the following feature selection techniques were used: 1) filter (chi-square, ANOVA); 2) wrapper (forward selection, backward elimination, recursive feature elimination); 3) embedded (decision tree based).

Filter Method (Chi-Square)

This method of feature selection is used when both the variables in the dataset are categorical. In the context of feature selection, the chi-square method is often used to assess the relationship between each predictor variable and the target variable (in this case, the disease outcome).

According to the Chi-Square test, the selected features below are a subset of variables that show a significant association with the disease outcome. These features have been identified as potentially relevant for predicting the outcome of the disease. It suggests that these variables may provide valuable information for predicting or understanding the disease.

Figure 8: Chi-Square test selected features

```
Selected Features:
Index(['pain_behind_the_eyes', 'throat_irritation', 'redness_of_eyes',
       'sinus_pressure', 'runny_nose', 'congestion', 'enlarged_thyroid',
       'brittle_nails', 'swollen_extremeties', 'slurred_speech',
       'loss_of_smell', 'increased_appetite', 'polyuria', 'rusty_sputum',
       'receiving_blood_transfusion', 'receiving_unsterile_injections', 'coma',
       'stomach_bleeding', 'blood_in_sputum', 'palpitations'],
      dtype='object')
```

Filter Method (ANOVA)

ANOVA method is used when one variable is categorical and the other is continuous. Under this method, univariate feature selection was carried out using the ANOVA F-value as the scoring function, and the top 20 features were selected based on this score. The features selected through ANOVA, were used in the following models and it was observed that the Accuracy remained same as compared to Recursive Elimination as conducted during the initial stages of the coding.

Table 4: Model and Accuracy

Model	Accuracy
Random Forest	0.98
Logistic Regression	0.9761904761904762
Support Vector Classifier	1.0

Figure 9: ANOVA F-value test selected features

```
Selected Features:
Index(['throat_irritation', 'redness_of_eyes', 'sinus_pressure', 'runny_nose',
       'congestion', 'enlarged_thyroid', 'brittle_nails',
       'swollen_extremeties', 'slurred_speech', 'loss_of_smell',
       'abnormal_menstruation', 'increased_appetite', 'polyuria',
       'rusty_sputum', 'receiving_blood_transfusion',
       'receiving_unsterile_injections', 'coma', 'stomach_bleeding',
       'blood_in_sputum', 'palpitations'],
      dtype='object')
```

Wrapper method (Forward selection)

Forward Selection begins with empty set of features and with each iteration it keeps adding on features and evaluates. This method was applied but it was too time consuming which proves to be a disadvantage while building ML model. These are the selected features under forward selection:

Figure 10: Forward Selection test selected features

```
Selected Features:
Index(['chills', 'sweating', 'nausea', 'mild_fever', 'malaise'], dtype='object')
```

Wrapper method (Backward elimination)

Backward elimination begins with all the features, with each iteration removing the least significant features. Below features have been identified as potentially relevant for predicting the outcome of the disease:

Figure 11: Backward Elimination test selected features

```
Selected Features:
Index(['itching', 'skin_rash', 'chills', 'joint_pain', 'vomiting', 'fatigue',
       'high_fever', 'headache', 'nausea', 'loss_of_appetite',
       'abdominal_pain', 'diarrhoea', 'chest_pain', 'loss_of_balance',
       'irritability'],
      dtype='object')
```

Wrapper method (Recursive feature elimination)

Recursive Feature Elimination (RFE) is a feature selection method that uses a machine learning model (in this case, Logistic Regression) to recursively eliminate less important features until a specified number of features (specified by n_features_to_select) is obtained to assure peak performance (Kelley, 2023). It will recursively eliminate less important features until 20 features are selected based on their importance according to the logistic regression model.

Figure 12: Recursive feature elimination test selected features

```
Selected Features:
Index(['itching', 'skin_rash', 'chills', 'joint_pain', 'vomiting', 'fatigue',
       'lethargy', 'cough', 'high_fever', 'sweating', 'headache',
       'yellowish_skin', 'nausea', 'loss_of_appetite', 'abdominal_pain',
       'diarrhoea', 'yellowing_of_eyes', 'chest_pain', 'excessive_hunger',
       'irritability'],
      dtype='object')
```

Embedded Method

In this study, decision tree based method for feature selection has been applied. Models assign importance scores to each feature based on how much they contribute to reducing error in the tree. Features with higher importance scores are considered more influential in the model's decision-making process. First threshold value was set to 0.25, which gave a null list of features. As, the threshold was too high, it was decreased to 0.01 and the following features were obtained:

Figure 13: Embedded Method – Decision Tree selected features

```
Selected Features:
Index(['chills', 'spotting_ urination', 'fatigue', 'weight_loss', 'cough',
       'high_fever', 'dark_urine', 'pain_behind_the_eyes', 'diarrhoea',
       'mild_fever', 'yellowing_of_eyes', 'swelling_of_stomach', 'phlegm',
       'chest_pain', 'bloody_stool', 'dizziness', 'bruising',
       'swollen_extremeties', 'excessive_hunger', 'knee_pain',
       'muscle_weakness', 'unsteadiness', 'loss_of_smell',
       'continuous_feel_of_urine', 'passage_of_gases', 'muscle_pain',
       'altered_sensorium', 'red_spots_over_body', 'abnormal_menstruation',
       'dischromic _patches', 'increased_appetite', 'family_history',
       'mucoid_sputum', 'rusty_sputum', 'lack_of_concentration',
       'receiving_unsterile_injections', 'coma', 'palpitations', 'blackheads',
       'small_dents_in_nails', 'yellow_crust_ooze'],
      dtype='object')
```

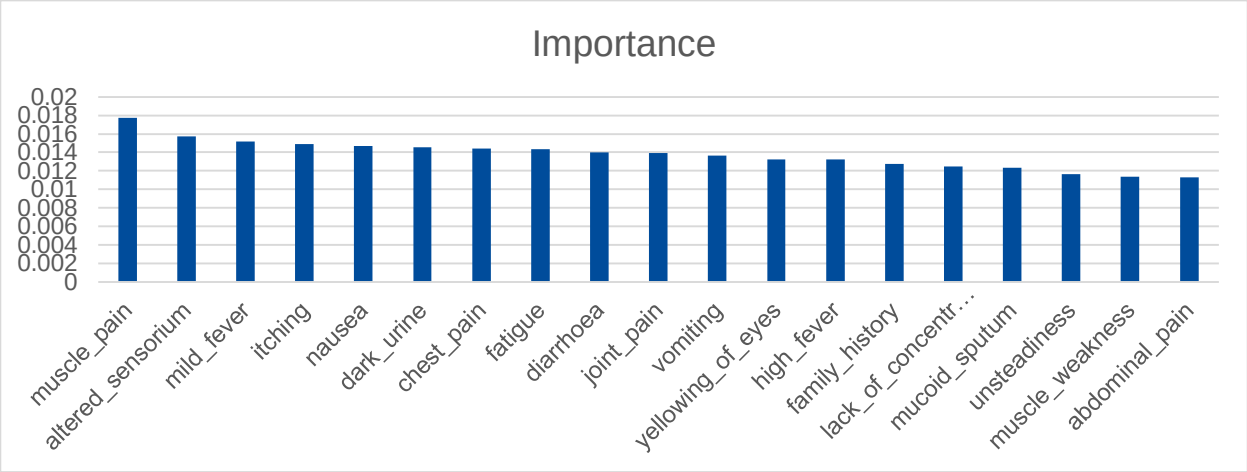
Recursive feature elimination (RFE) has been used in the code because of its advantages over other techniques. RFE is a wrapper-based feature selection algorithm that works by iteratively removing the least important features from a dataset until the desired number of features is reached.

Feature Importance

Feature importance is a measure of the relative importance of each feature in predicting the target variable in a machine learning model. The higher the importance value, the more influential the feature is in making predictions.

In this study, each feature is assigned an importance value. For example, "muscle_pain" has an importance value of 0.019984, "joint_pain" has an importance value of 0.016964, and so on. These values indicate the significance of each feature in the predictive model. The following graph shows important features in descending order:

Graph 7: Feature Importance



Features with higher importances are considered more relevant and contribute more to the overall prediction. On the other hand, features with lower importances have less influence on the predictions. It helps defining symptom's association with the target variable in the train dataset.

Modeling Algorithms

LOGISTIC REGRESSION

Logistic Regression was the first classification algorithm used for predictive modelling. It is commonly used algorithm in disease prediction models as it estimates the probability of an individual having a certain disease based on the given input variable (symptom). A Logistic Regression model was created through SKLearn linear_model function and other functions like `accuracy_score` and `classification_report` for evaluating the model's performance.

RANDOM FOREST

Next, Random Forest algorithm was used for predictive modelling. The intention was to use decision tree model but instead I went ahead with Random Forest because decision tree tend to overfit the training data, meaning they can capture irrelevant pattern. Random Forest help mitigate overfitting by creating an ensemble of multiple decision trees and aggregating their prediction.

As this dataset is a high dimensional data, decision tree often struggles with high dimensional data where the number of features are large. Whereas Random Forest can handle this dataset more effectively by randomly selecting a subset of features at each node, ensuring all the features have an equal chance to contribute in the prediction. Hence, it makes more accurate prediction.

While building Random Forest Classifier, n_estimators were taken equals to 100 (decision trees). In order to create a classification, report I used default parameters of precision, recall and F-1 score but it was giving an error that it cannot run on multiclass variable. Therefore, parameters such as pos_label='positive' and average='micro' were added to get the classification report.

K NEAREST NEIGHBOR (KNN)

The K-Nearest Neighbors (KNN) algorithm is used in this disease prediction model as diseases can often have intricate relationships with multiple symptoms, and KNN can be flexible in capturing these patterns. KNN is useful in this study as the data distribution is complex (Uddin et al., 2022).

In disease prediction, the distribution of symptoms and their relationship with the disease outcome can vary widely, and KNN can adapt to different distributions. During inference, the computational cost mainly depends on the number of instances (training data) rather than the complexity of the mode. This can be advantageous in disease prediction models where quick and efficient predictions are desired (Uddin et al., 2022).

In the code, KNN classifier is initialised by creating an instance of n_neighbors=3, which signifies that the number of neighbors to be considered are set as 3.

SUPPORT VECTOR MACHINE CLASSIFIER

Support Vector Machines (SVM) is capable of capturing complex, nonlinear relationships between features and the target variable (prognosis). In the code, default parameters were used and by default, `SVC()` in scikit-learn uses the Radial Basis Function (RBF) kernel.

SVMs perform well even when the number of features is high. It is useful in this disease prediction study, where the dataset contains a large number of symptoms. Another advantage of SVM is that it is less prone to overfitting in large dataset like the one used in this study.

GAUSSIAN NAÏVE BAYES

Gaussian Naive Bayes assumes that the features (symptoms in this study) are conditionally independent given the class label. In disease prediction, it is often reasonable to assume that the presence or absence of different symptoms may be independent of each other. It performs well even when the number of features is large compared to the number of samples. This makes it particularly useful when dealing with a high-dimensional data, such as medical data in this study, where the number of potential predictors may be large.

Gaussian Naive Bayes results are interpreted by calculating class probabilities based on Bayes' theorem. It can provide insights into the likelihood of a patient having a certain disease based on the observed symptoms or risk factors.

Results

First Iteration

The algorithms were all ran using their default settings in the first iteration.

Logistic Regression

The classification report provides detailed metrics such as precision, recall, F1-score, and support for each class in the classification task.

The accuracy score is 0.9761904761904762, which indicates that the model correctly predicted the class labels for 97.62% of the instances in the dataset.

In this case, most classes have a precision of 1.00, indicating high precision for those classes but for chicken pox precision is 0.50. A precision score of 0.50 suggests that the model's predictions for the "Chicken pox" class are not as reliable or accurate compared to other classes in the classification report.

Similar to precision, most classes have a recall of 1.00, indicating high recall for those classes. In the case of "Fungal infection," the model had a lower recall compared to other classes, indicating that it may have more difficulty accurately predicting instances of this class.

Most classes in the report have an F1-score of 1.00, indicating high accuracy. A F1-score of 0.67 suggests that the model achieved moderate accuracy in identifying instances of fungal infection.

Random Forest

The model achieved an accuracy of approximately 97.6%. The precision, recall, and F1-scores for most classes are 1.00, indicating high performance. However, for some classes (e.g., Fungal infection and Impetigo), the F1-score is slightly lower, suggesting that the model may struggle to predict those classes accurately.

K-Nearest Neighbor

In this case, the model achieved perfect performance with an accuracy, precision, recall, and F1-score of 1.0 for all classes. This means that the model made no errors in predicting the classes and correctly classified all samples in the test set, demonstrating good performance on the disease prediction task.

In summary, the KNN model with `n_neighbors=3` achieved perfect accuracy and performance, accurately predicting all classes in the disease prediction task.

Support Vector Classifier and Gaussian Naïve Bayes

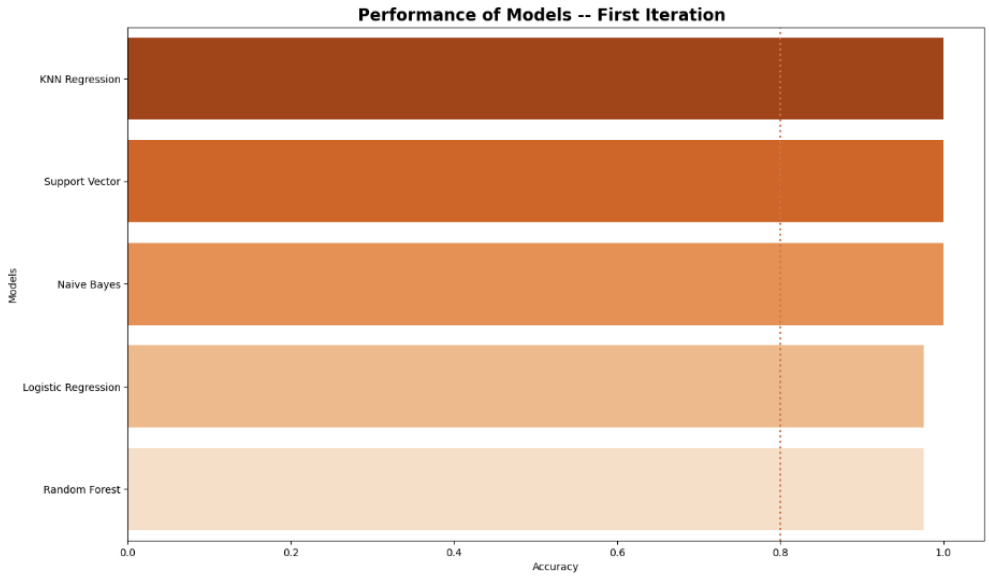
The results from the classification report and evaluation metrics for both the models gave an accuracy of 1.0 signifying good performance on the disease prediction dataset. A precision of 1.0 indicates that there were no false positives for any class, meaning all positive predictions were correct. A recall of 1.0 indicates that there were no false negatives for any class, meaning all positive instances were correctly identified. An F1-score of 1.0 indicates perfect precision and recall for all classes.

From the evaluation metric results of the first iteration of each of the models, it can be observed that K-Nearest Neighbor, Support Vector and Naive Bayes algorithms performed the best with 100% accuracy, while Logistic Regression and Random Forest had 97.00% accuracy.

Table 5: Evaluation metrics – 1st iteration of modeling

Model	Accuracy	Precision	Recall	F1 Score
KNN Regression	1.00000	1.00000	1.00000	1.00000
Support Vector	1.00000	1.00000	1.00000	1.00000
Naïve Bayes	1.00000	1.00000	1.00000	1.00000
Logistic Regression	0.97619	0.99000	0.99000	0.98000
Random Forest	0.97619	0.97619	0.97619	0.97619

Figure 14: Each algorithm's performance after 1st iteration by accuracy



Hyper Parameters

The objective is to select the most optimal collection of parameters to control the algorithm to provide the best performance using hyper-tunning. A grid search methodology (Pandian, 2022b) is applied to all five algorithms. The best parameters from hypertuning were chosen in the second modelling iteration and were as follows:

Table 6: Hyperparameters for all five algorithms

Algorithm	Hyper Parameters:
KNN Regression	{'n_neighbors': 1, 'p': 1, 'weights': 'uniform'}
Support Vector	{'C': 0.001, 'gamma': 0.001, 'kernel': 'linear'}
Naïve Bayes	{'var_smoothing': 1e-09}
Logistic Regression	{'C': 0.1, 'penalty': 'l2'}
Random Forest	{'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}

Second Iteration

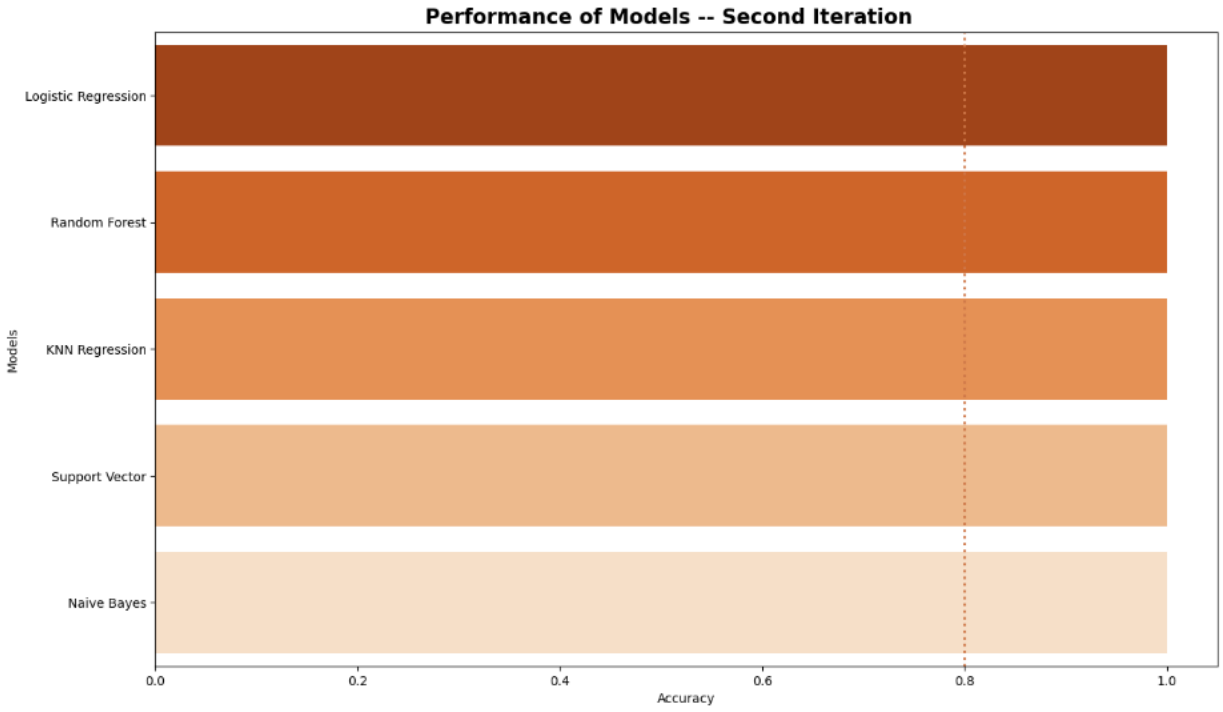
In the second iteration, the algorithms were all ran again with the above mentioned hyper tuned parameters.

From evaluation metrics after the second iteration it can be seen that hyper-tunning increased the performance of Logistic Regression and Random Forest to 100%. Accuracy remained the same for the other models as the initial hyper parameters used in the model were already close to optimal values and further tuning may not lead to a significant change in accuracy.

Table 7: Evaluation metrics – 2nd iteration of modeling

Model	Accuracy	Precision	Recall	F1 Score
KNN Regression	1.00000	1.00000	1.00000	1.00000
Support Vector	1.00000	1.00000	1.00000	1.00000
Naïve Bayes	1.00000	1.00000	1.00000	1.00000
Logistic Regression	1.00000	1.00000	1.00000	1.00000
Random Forest	1.00000	1.00000	1.00000	1.00000

Figure 15: Each algorithm's performance after 2nd iteration by accuracy

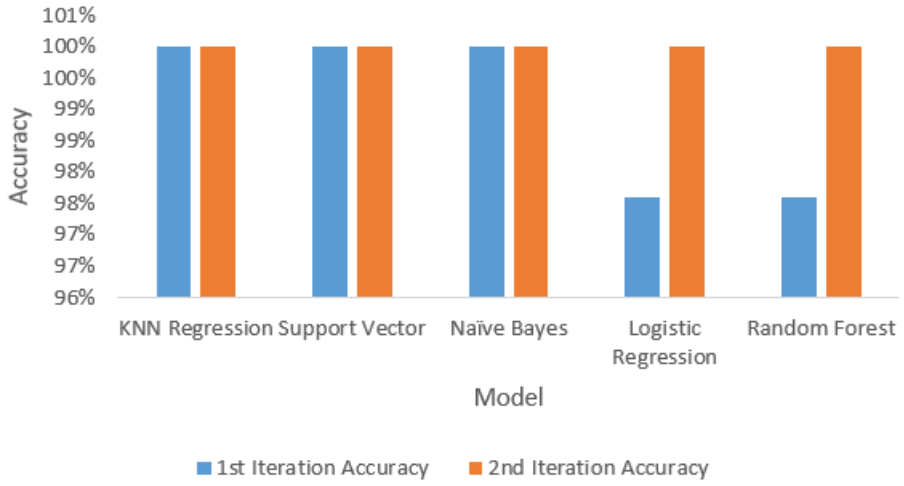


Comparing the two iterations of modeling, there was no change in accuracy after hyper-tuning for the K-Nearest Neighbor, Support Vector and Naïve Bayes. Both Logistic Regression and Random Forest had a 2.39% increase in accuracy.

Table 8: Accuracy Comparison

Model	1st Iteration Accuracy	2nd Iteration Accuracy	Difference in Accuracy
KNN Regression	100%	100%	0%
Support Vector	100%	100%	0%
Naïve Bayes	100%	100%	0%
Logistic Regression	97.61%	100%	2.39%
Random Forest	97.61%	100%	2.39%

Graph 8: Performance of algorithm after both iterations



Cross Validation

Leave One Out Cross Validation

Leave One Out Cross Validation was tested on Logistic Regression. The error value of 0.2 obtained from the leave-one-out cross-validation indicates the average error rate of the Logistic Regression model on this dataset. In this case, the error rate represents the misclassification rate of the model, where a lower value indicates better performance.

It indicates that the model misclassified approximately 20% of the instances in the dataset when evaluated using leave-one-out cross-validation. This method is useful when working with small datasets where the number of samples is limited. It does not seem to be of much use in this dataset as the number of samples was significantly high.

Stratified K-Fold Cross Validation

K-Fold Cross-Validation was tested on Support Vector algorithm. In the context of K-Fold cross-validation, k=5 and k=10 yielded same results indicating that the model achieved perfect accuracy (1.0) on each fold of the cross-validation. This means that the model correctly predicted the disease outcome for all instances in each fold. The standard deviation of 0.0 suggests that the accuracy scores across the folds are identical or very close to each other, as there is no variability in the accuracy values.

Overall, an accuracy of 1.0 with a standard deviation of 0.0 in k-fold cross-validation is an ideal scenario, suggesting that the model is accurately capturing the relationship between the symptoms and the disease prognosis.

Hold Out Cross Validation

The holdout cross-validation accuracy is 0.33333333333333337. This was tested on Logistic Regression. This means that the model was able to correctly predict 33.33% of the test data. In holdout cross-validation, the dataset is randomly split into two parts: a training set and a test set. The training set is used to train the model, and the test set is used to evaluate the model's accuracy. Accuracy in this case is low as the test dataset was small enough to cross validify through this method.

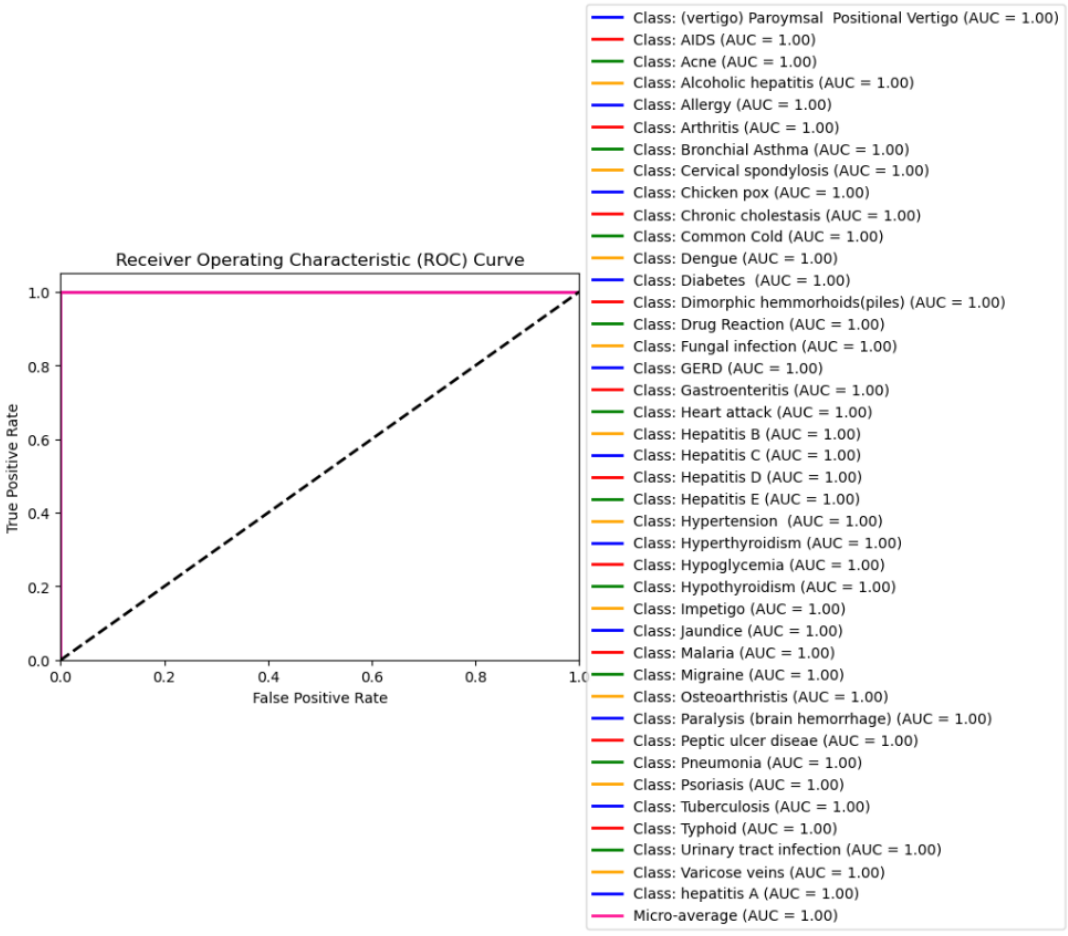
Receiver Operating Characteristic Curve (ROC)

The ROC curve is created by plotting the true positive rate against the false positive rate at various classification thresholds. The AUC-ROC represents the overall performance of the model in distinguishing between the two classes.

As Support Vector has the highest accuracy score, a ROC Curve graph was plotted to show the performance of Support Vector model at all classification thresholds.

The AUC-ROC score of 1.0 signifies excellent performance, indicating that the model is highly effective in predicting the presence or absence of the disease.

Figure 16: ROC Curve of Support Vector



Conclusions

The main goal of this study was to predict the disease on the basis of the symptoms. This paper provides an overview of various machine learning algorithms such as KNN, Logistic Regression, Random Forest, SVM and Naïve Bayes used in disease prediction models. The performance of all the algorithms are compared and it was seen that algorithms such as KNN, Support Vector and Naïve Bayes provide an accuracy with available attributes in the dataset that is higher than that of the Logistic Regression and Random Forest.

Research Questions

Can we accurately predict disease using machine learning?

The results of the modeling confirm that we can accurately predict diseases with the given symptoms in this dataset. After the first iteration, the best performing algorithm were K-Nearest Neighbor, Support Vector and Naïve Bayes with a 100% accuracy. With hyper-tunned parameters, Logistic Regression and Random Forest yielded an accuracy of 100% while the others remained the same as they already had optimal results prior to hyper-tunning.

What method of cross validation evaluation will be employed?

Leave one out and Hold out cross validation methods were used on Logistic Regression as initially the accuracy for Logistic Regression was low compared to the other models. Moreover, Stratified K-Fold cross validation was used on SVM as its accuracy was 100% so to cross validify the stability of the model with best accuracy it seems to be a necessity. In this case, K-Fold Cross Validation seems to be the optimal choice.

Can the ML algorithms' settings be tuned for the optimal performance?

From evaluation metrics after the second iteration it can be seen that hyper-tunning increased the performance of Logistic Regression and Random Forest to 100%. Accuracy remained the same for the other models as the initial hyper parameters used in the model were already close to optimal values and further tuning may not lead to a significant change in accuracy.

Which machine learning algorithms are most effective in producing reliable results?

Due to the size of the dataset, there were delays while running the models in terms of processing time. Time taken to run Random Forest model was too much so it was not the appropriate model as increasing the number of decision trees through hyperparameter running increases run-time. Run time was comparatively less for Support Vector and K-Nearest Neighbor.

Which feature selection technique will be appropriate for this dataset?

Recursive Feature Elimination (RFE) feature selection method that recursively eliminates less important features until features are selected based on their importance according to the logistic regression model.

References

Kanakaraddi, S. G., Gull, K. C., Bali, J., Chikaraddi, A. K., & Giraddi, S. (2021b). Disease Prediction Using Data Mining and Machine Learning Techniques. In *Lecture notes on data engineering and communications technologies* (pp. 71–92). Springer International Publishing. https://doi.org/10.1007/978-981-16-0538-3_4

Grampurohit, S., & Sagarnal, C. (2020). Disease Prediction using Machine Learning Algorithms. In *2020 International Conference for Emerging Technology (INCET)*. https://doi.org/10.1109/incet49848.2020.9154130

Disease. (n.d.). https://impact.dbmi.columbia.edu/~friedma/Projects/DiseaseSymptomKB/index.html

Mahata, S., Kapadiya, Y. B., Kushwaha, V., Joshi, V., & Farooqui, Y. (2023). Disease Prediction and Treatment Recommendation Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, *11*(3), 1232–1237. https://doi.org/10.22214/ijraset.2023.49641

Arumugam, K., Naved, M., Shinde, P. P., Leiva-Chauca, O., Huaman-Osorio, A., & Gonzales-Yanac, T. (2021). Multiple disease prediction using Machine learning algorithms. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2021.07.361

Kolte, A., Mahitha, B., & Raju, N. V. G. (2019). *Stratification of Parkinson Disease using python scikit-learn ML library*. https://doi.org/10.1109/icese46178.2019.9194627

Radhika, S., Shree, S. R., Divyadharsini, V. R., & Ranjitha, A. (2020). *Symptoms Based Disease Prediction Using Decision Tree and Electronic Health Record Analysis*. European Journal of Molecular & Clinical Medicine. https://ejmcm.com/article_1944_cb7aaa34894c921618817c5c40cdaf5d.pdf

Ferjani, M. (2020, December 16). *Disease prediction using machine learning* . Research Gate. https://www.researchgate.net/profile/Marouane-Ferjani/publication/347381005_Disease_Prediction_Using_Machine_Learning/links/5fda5556299bf1408816daa4/Disease-Prediction-Using-Machine-Learning.pdf

Uddin, S., Haque, I., Lu, H., Moni, M. A., & Gide, E. (2022b). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, *12*(1). https://doi.org/10.1038/s41598-022-10358-x

Kelley, K. (2023). Recursive Feature Elimination: What It Is and Why It Matters. *Simplilearn.com*. https://www.simplilearn.com/recursive-feature-elimination-article

Matthews Correlation Coefficient. (2022). *encyclopedia.pub*. https://encyclopedia.pub/entry/35211

Pandian, S. (2022). A Comprehensive Guide on Hyperparameter Tuning and its Techniques. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2022/02/a-comprehensive-guide-on-hyperparameter-tuning-and-its-techniques/

Zach. (2020). What is a Brier Score? *Statology*. https://www.statology.org/brier-score/

Gupta, A. (2023). Feature selection techniques in Machine Learning (Updated 2023). *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/

Feature importance | Machine Learning in the Elastic Stack [8.8] | Elastic. (n.d.). Elastic. https://www.elastic.co/guide/en/machine-learning/current/ml-feature-importance.html

Comotto, F. (2022, January 15). Evaluation metrics: leave your comfort zone and try MCC and Brier Score. *Medium*. https://towardsdatascience.com/evaluation-metrics-leave-your-comfort-zone-and-try-mcc-and-brier-score-86307fb1236a