Performance Comparison of ML Algorithms for Sustainable Smart Health Systems

C. N. Sujatha1, K. Swaraja2*, C. Bharath Kumar1, Sai Roshit1, Sucheet1, Sandeep3, M.N. Narsaiah4

1 ECE Department, Sreenidhi Institute of Science and Technology, Hyderabad, India.
2 ECE Department, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India.
3 Uttarakhand Institute of Technology, Uttarakhand University, Dehradun, India.
4 KG Reddy College of Engineering & Technology, Hyderabad, India.

Abstract. Disease prognosis holds immense significance in healthcare due to its potential to greatly improve patient outcomes through early and precise diagnosis. Machine learning (ML) algorithms provide a robust avenue for disease prediction, employing patient data analysis to detect intricate patterns of specific ailments. Machine learning algorithms adeptly handle intricate and extensive datasets, uncovering latent patterns often eluding human observation. By considering diverse symptoms and their permutations, ML models yield precise forecasts concerning the probability of distinct diseases.

1 Introduction

The significance of sustainability is becoming more and more crucial as healthcare systems continue to change. The findings and recommendations of this work provide guidance for healthcare professionals, decision-makers, and technologists who want to promote innovation while putting an emphasis on resource efficiency and reducing their environmental impact.

Predicting diseases is crucial in healthcare, helping with early diagnosis and better patient outcomes. With electronic health records and large patient data, machine learning (ML) models have been developed for disease prediction based on symptoms. The Decision Tree approach predicts ailments handling both categorical and numerical traits, Decision Trees encapsulate intricate symptom interactions. Techniques like pruning and ensemble strategies curb overfitting.

Random Forest (RF) merges numerous Decision Trees (DT) to amplify prediction precision, addressing individual Tree limitations by mitigating overfitting and enhancing generalization. It erects multiple trees using random feature and data subsets. Aggregating predictions, the

*corresponding author: kswaraja@gmail.com
2 Literature survey
3 Methodology
NB is considered as a probabilistic method that assumes feature independence. It computes disease probability based on symptoms applying the Baye’s theorem. Naive Bayes classifiers are simple, efficient, and apt for calculating diseases. They operate on sizable datasets, offering swift predictions.

LR notably predicts disease presence or absence. It models symptom-disease connections using a logistic function. Logistic Regression gauges binary outcome probability from input traits. It excels when symptom-disease links are fairly linear. The models provide intelligible coefficients delineating feature influence on disease probability.

SVM represents a potent algorithm for classification duties, encompassing disease prognosis. It diligently pursues optimal separation of distinct classes within a high-dimensional feature realm. Symptoms translate into feature vectors, and SVM discerns a prime hyperplane for precision classification. SVM proficiently manages linear and non-linear symptom-disease connections via diverse kernels. Yet, SVM's computational demands might necessitate judicious parameter calibration.

3.1 Implementation

Step 1: Data collection and preprocessing

The operationalization of disease prediction necessitates the accumulation of a dataset encompassing symptomatic manifestations alongside their corresponding maladies. This reservoir of information can be sourced from diverse outlets, encompassing medical institutions, scientific periodicals, and public health entities. Upon the culmination of data compilation, the amassed dataset mandates preparatory actions. This preparatory phase engulfs a gamut of tasks, encompassing data cleansing, management of absent values, and the conversion of data into the mandated structure.

Step 2: Feature Engineering

Feature engineering involves the process of choosing and modifying attributes within a dataset. In the context of disease prediction, symptoms serve as these attributes. The selection of attributes is based on their relevance in predicting the disease. Once chosen, these attributes need to be transformed into a format suitable for utilization by machine learning algorithms. These techniques aid in reducing the complexity of the dataset and enhance the model's performance by focusing on the most informative symptoms.

Step 3: Sustainable Model Selection

Selecting the sustainable algorithm is based on various metrics such as the type of the data, the size of the datasets and the values corresponding and the accuracy required for the disease prediction.

Step 4: Model Training

After selecting the referring algorithm, train the model on the dataset. Then the algorithm adapts the relationship between the symptoms and the corresponding diseases.
Step 5: Model Evaluation

Evaluate its performance on test set. ML creditable frameworks is calculated using evaluation metrics includes all the classifiers and metric value estimators.

Step 6: Model Execution

Then algorithm is trained it can be executed in many ways. Then create an app on web which accepts symptoms as input, predicts the disease using the trained model.

4 Results & Discussion

The dataset utilized for disease prediction in this framework encompasses a wide spectrum of symptoms and their corresponding diseases. It is amassed from various sources such as medical records, journals, and health organizations. Each entry in the dataset comprises a unique amalgamation of symptoms and their associated diseases, and this dataset serves as the foundation for training and evaluating the machine learning algorithm's disease prediction capabilities.

The dataset comprises a total of 41 diseases and 133 symptoms. Each disease is represented as a row, and each symptom as a column. The dataset is divided into two parts: a testing dataset with 41 rows and 133 columns, and a training dataset with a substantial 1,048,575 rows and 133 columns. Each cell in the training dataset may contain binary data 1 or 0 indicating the presence 1 or absence 0 of a symptom for a specific disease. This dataset appears to be a significant collection of symptoms and diseases, with the training data intended for model training and the testing data for evaluating the trained model predictive performance. The testing dataset contains 41 rows, corresponding to the 41 diseases with 133 symptoms as columns. Similar to the training dataset, each cell in the testing dataset may contain binary data (1 or 0), indicating the presence (1) or absence (0) of a symptom for a specific disease.

Accuracy serves as the metric denoting the correctness of prognostications, derived by appraising the quotient of precise predictions against the aggregate of predictions rendered. This evaluative tool assumes paramount importance in the scrutiny of machine learning algorithms.

The mathematical equation to calculate Accuracy is given in Eq(1).

\[
\text{Accuracy} = \frac{\text{True P} + \text{True N}}{\text{True P} + \text{True N} + \text{False P} + \text{False N}}
\]  

(1)

Precision is the degree of exactness inherent in predictions, ascertained through the computation of the proportion between accurate predictions and the entirety of predictions made. This metric stands as a pivotal yardstick for the appraisal of machine learning algorithms. However, its utility might dwindle when applied to imbalanced datasets where the scale of affirmative and negatory instances undergoes substantial fluctuations. Table 1 provides the performance metrics of different ML models.

The mathematical equation to calculate Precision is given in Eq(2).

\[
\text{Precision} = \frac{\text{True P}}{\text{True P} + \text{False P}}
\]  

(2)
Table 1. Performance Metrics of ML models

<table>
<thead>
<tr>
<th>Name of the Classifier</th>
<th>Accuracy (Train and Validation data)</th>
<th>Accuracy (Test data)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.9857</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>98.58</td>
<td>98.58</td>
</tr>
<tr>
<td>DT</td>
<td>0.9680</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>96.81</td>
<td>96.81</td>
</tr>
<tr>
<td>NB</td>
<td>0.9863</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>98.64</td>
<td>98.64</td>
</tr>
<tr>
<td>RF</td>
<td>0.7081</td>
<td>77.506</td>
</tr>
<tr>
<td></td>
<td>70.82</td>
<td>70.82</td>
</tr>
<tr>
<td>LR</td>
<td>0.9858</td>
<td>100.0</td>
</tr>
<tr>
<td></td>
<td>98.58</td>
<td>98.58</td>
</tr>
</tbody>
</table>

Recall given in Eq(3) is identified as the sensitivity rate, quantifies the proportion of true positive prognostications in relation to the comprehensive count of veritable affirmative instances. It exemplifies the algorithm's adeptness in accurately discerning affirmative cases within the positive forecasts.

F1 score given in Eq(4) calculates FP’s and TP’s and FN’s, making it useful when the dataset is not balanced. If the f1 score is higher, then precision and recall are in better balance algorithmic value state.

The generation of the confusion matrix necessitates the alignment of actual disease labels with their corresponding predictions. The rows within the matrix embody actual disease categories, while columns represent the predicted disease outcomes. By comparing these values, we obtain insights into the model's predictive performance in diagnosing diseases based on symptom data. The performance of Naive Bayes algorithm in terms of precision, recall, and other evaluative metrics can be extrapolated from this matrix. So the confusion matrix is an instrument used to evaluate the performance of machine learning algorithms,
reflects the predictive efficacy of the Naive Bayes algorithm, which boasts an accuracy of 0.9863. The confusion matrix portrays the comparison between actual and predicted outcomes. The vertical and horizontal components of the matrix symbolize the actual and predicted disease categories, respectively.

![Confusion Matrix for Naive Bayes Classifier on Test Data]

Fig.1. Confusion matrix - Naive Bayes having the Maximum Accuracy output values

Figure.1 is based on the algorithm out of all the five (NB) representing the Maximum accuracy values compared to the rest of the techniques used. It has the best overall score among all the classifiers used for prediction of disease and its corresponding symptoms. Table 2 provides the accuracy values of different classifiers. Max Accuracy is obtained for Naive Bayes Classifier as 0.9863.

<table>
<thead>
<tr>
<th>Classifier Name</th>
<th>Accuracy Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.9854</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>0.9671</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.9863</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.7081</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.9857</td>
</tr>
</tbody>
</table>
5 Conclusion & Future Scope

Besides picking the right algorithm, it's also important to use the right tools. This study used popular Python libraries like Pandas, NumPy, and Scikit-learn. These tools helped with getting the data ready, training the models, and making a user-friendly interface. The system's interface lets users put in symptoms and choose an algorithm to predict possible diseases. They used Tkinter, which made it easy to create buttons, labels, and text boxes for the user interface. To sum it up, making disease prediction systems with machine learning and Python tools can be really useful for doctors and healthcare workers. The system's predictions can help them make better decisions about patient care. Future research can look at other machine learning methods and include more details like patient backgrounds, medical history, and environment factors to make disease predictions even better.

References


