HIGH SCHOOL EDITION
Journal of Student Research
Volume 13 Issue 1 (2024)

Disease Prediction Using Machine Learning Methods

Christina Zhuang[1] and Ramin Ramezani[#]

[1]Newport High School, USA
[#]Advisor

ABSTRACT

A visit to the doctor's office usually starts with the nurse collecting patient symptoms, health information, and necessary lab tests. All the information will be presented to the doctor, and the doctor may collect additional information in order to do the right diagnosis. The doctor's brain is like a complicated machine capable of quick processing of the information, relating it to previous patients, and mapping the information to diagnoses. This process resembles much to how machine learning works. In this article, we explore how machine learning could help predict different diseases and facilitate a doctor's diagnosis. In particular, our study focuses on unbalanced, multiclass classification problems.

Introduction

Some diseases show common symptoms, for example, common cold and pollen allergy could both cause runny nose, sore throat, etc. Doctors rely on additional information to differentiate them. Other diseases may take time to manifest themselves. For example, Rett Syndrome [2] may not show obvious symptoms until 6 months or more after birth. How to detect such disease early on is becoming a challenge. Fortunately, overtime, the healthcare workers have collected lots of helpful information about various diseases and their symptoms. If doctors can leverage the collective learnings and tips from other doctors and institutions, they can make better and earlier diagnoses. One challenge would be how to effectively consolidate the information from different sources and accurately map it to correct diagnoses. Another challenge is how to reduce human error and consistently deliver high-quality diagnosis. With the fast improvement of computation power and artificial intelligence, we wonder how machine learning techniques could help with this process and overcome the challenge. If these technologies could help improve the diagnosis accuracy by 1% or reduce the time it takes to reach a conclusion by 1%, millions of patients could get better or earlier treatments around the world!

Related Work

Machine learning based disease diagnosis has been an active research topic and has gained lots of traction since 2018 (Ahsan et al., 2022). Severe diseases like COVID-19, heart disease, diabetes, kidney disease, breast cancer, Alzheimer's, etc. have received more attention due to their severity and importance. Bermando et al. used Naive Bayes (NB) and Random Forest (RF) algorithms to predict coronary heart disease and achieved an 85% accuracy (Bemando et al., 2021). Acharya et al leverage CNN to predict different patterns of heartbeats and get accurate models on both balanced and unbalanced data (Acharya et al., 2017). Charleonnan et al. used publicly available datasets to evaluate four different ML algorithms: K-nearest neighbors (KNN), support vector machine (SVM), logistic regression (LR), and decision tree classifiers on kidney disease diagnosis and received the accuracy of 98.1%, 98.3%, 96.55%, and 94.8%, respectively (Charleonnan et al., 2016). Asri et al. applied

ML approaches such as SVM, DT, NB, and KNN on the Wisconsin Breast Cancer (WBC) datasets. According to their findings, SVM outperformed all other ML algorithms, obtaining an accuracy of 97.13% (Asri et al., 2016). Naz and Ahuja employed a variety of ML techniques, including artificial neural networks (ANN), NB, and DT, to analyze open-source PIMA Diabetes datasets with the best accuracy of approximately 98.07% (Naz & Ahuja, 2020). Neelaveni and Devasana proposed a model that can detect Alzheimer patients using SVM and DT and achieved an accuracy of 85% and 83% respectively (Neelaveni &Devasana, 2020). On the more recent COVID-19 diagnosis, Chen et al. proposed a UNet++ CNN model employing CT images from 51 COVID-19 and 82 non-COVID-19 patients and achieved an accuracy of 98.5% (Chen et al., 2020). On other disease prediction, Khan et al. employed CNN-based approaches such as VGG16 and VGG19 to classify multimodal Brain tumors and achieved more than 92% accuracy (Khan et al., 2020). These previous works mostly focused on predicting single disease and optimizing for higher model accuracy.

Approach

There had been lots of prior work on utilizing machine learning to do disease diagnosis. Most of the work focused on optimizing the model accuracy on a single class prediction. In reality, many data will have multiple classes, often with unbalanced data distribution, and some extent of noise. We will study and evaluate 7 common machine learning models in the setting of multiclass, unbalanced data. We will

- perform systematic evaluations of various models on disease prediction
- make informative comparisons between the models
- share recommendations on which algorithm/model is more suitable for the diagnosis task given time and accuracy constraints.

  We also keep in mind that the approaches should apply to other datasets and other disease types. We mainly use Jupyter notebook and the scikit-learn libraries to train/evaluate the models [3].

Dataset

The data set for this investigation is from Kaggle machine learning competition [4]. It contains 42 diseases with 132 symptom features for each patient. The data set has a training data file with 14760 entries and a validation data file with 2952 entries. This is a multi-class supervised learning problem. Some of the entries in the training set are random noise.

Data Cleaning

Upon checking the training data file, we found one column with no values. The column is not helpful for model training and we removed it upfront to save processing down the road.

Exploratory Data Analysis

Among the 133 columns, only one column (“prognosis”) is categorical, containing the disease names, all the other columns are numerical and have values 0 or 1.

Summary of the Dataset:

- Number of Entries (Rows): 14760
- Number of Columns: 133

Column Information:

- Column Names: *itching, skin_rash, nodal_skin_eruptions, continuous_sneezing, shivering, chills,* and more...
- Data Types: *132 columns with integer values, 1 column with object (prognosis)*
  The data set has no null values. So overall it is a well-formed and user-friendly dataset.

Data Distribution: As shown in Figure 1. The entries are pretty evenly distributed across the disease types. However, from each individual disease's point of view, the data is very unbalanced since on average the positive data is only about 2.4% of the total data set and the majority is negative data.

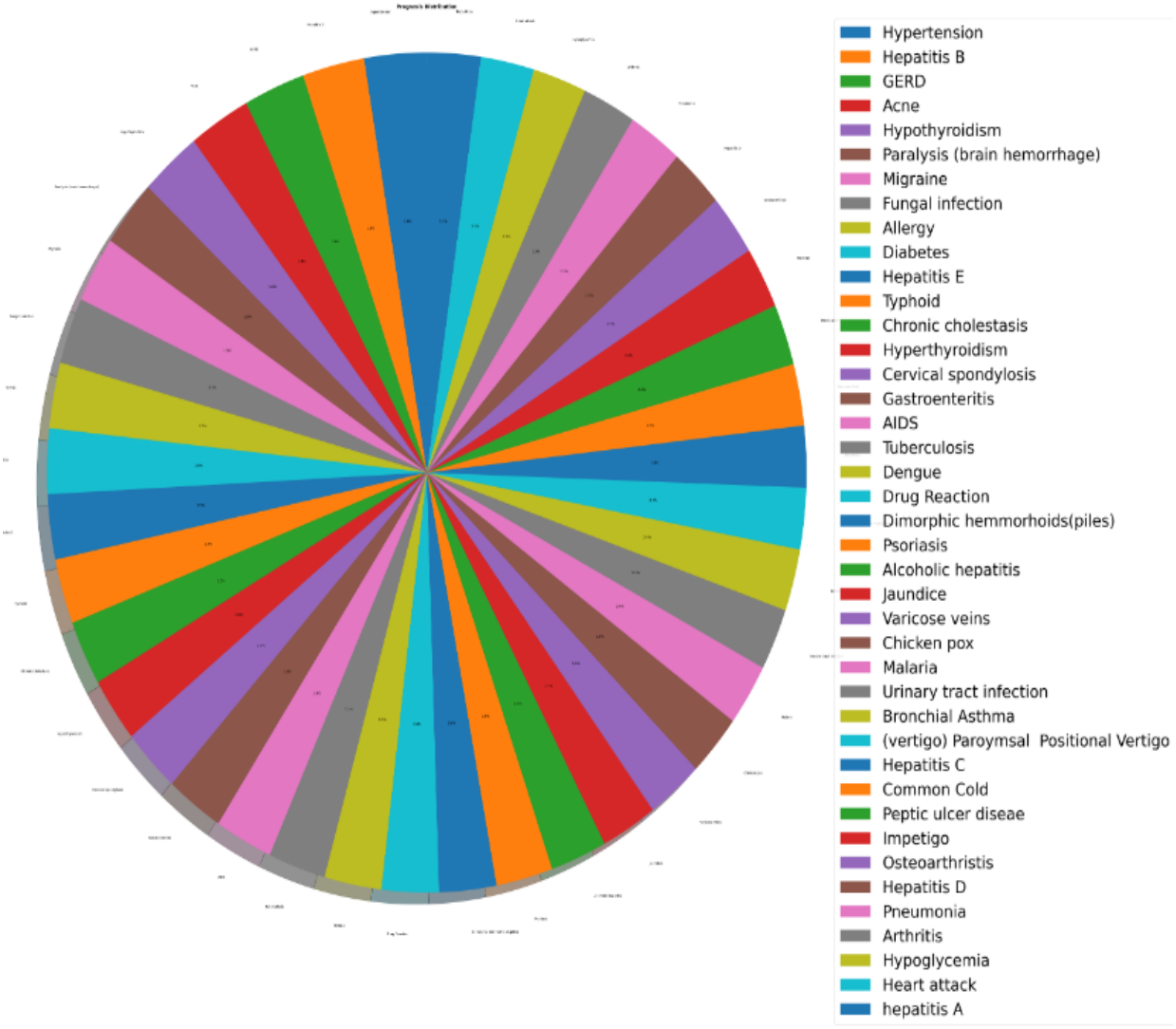

Figure 1. Training data distribution by disease type

Feature Correlation Matrix: This is to plot the heatmap of the features to see if any of the features/symptoms are correlated to each other.

We can see that while most features are independent of each other, some features do have high correlations, for example, the throat_irritation, redness_of_eyes, sinus_pressure,runny_nose, congestion features are almost perfectly positive correlated since they often co-occur in some diseases like "Common Cold".

Looking at the data points away from the diagonal, we can see quite some noise. It would be up to each model to be resilient to the noises.

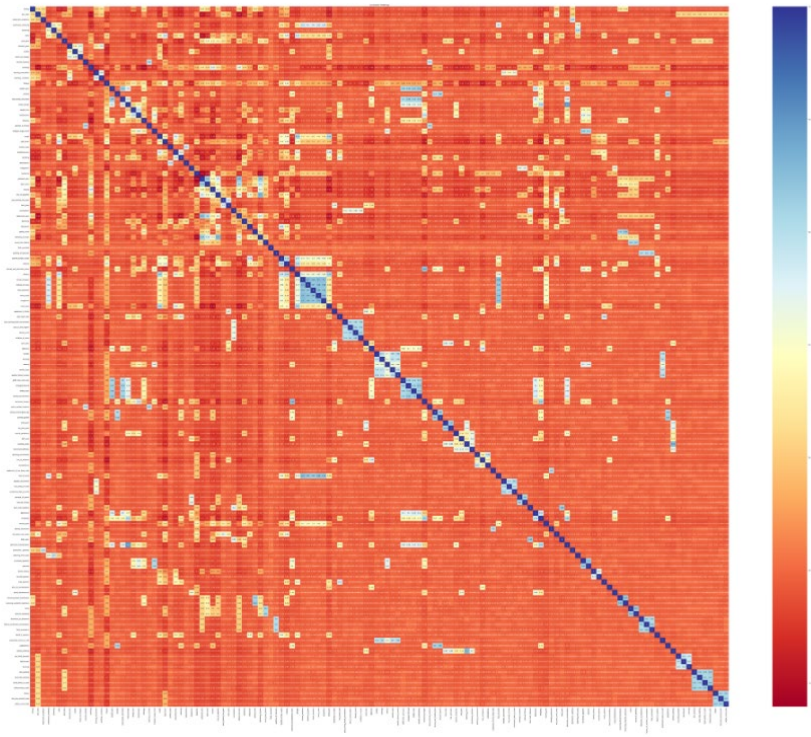

Figure 2. Training data feature correlation heat map for all 132 features

How is each feature related to the label? We have checked the correlation between any two features. But how is each feature related to the diagnosis (the label)? We can do cross tabulation between each feature and the class label.

Here are some of the example plots:

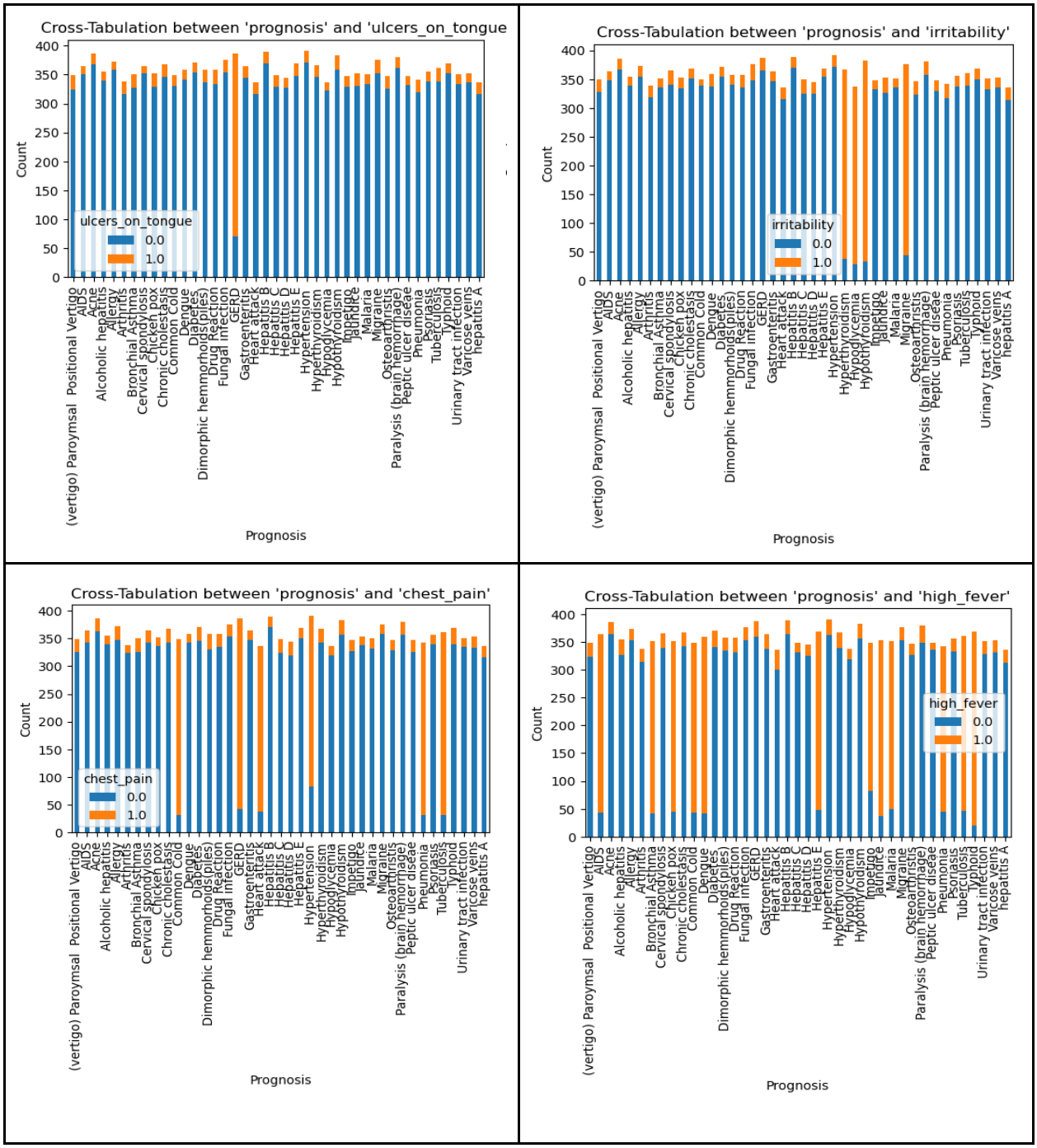

Figure 3. The correlation between the feature and the label. Example feature “ulcers_on_tongue” and “irritability”, “chest_pain”, “high_fever”

The cross tabulation can show us the correlation between each feature and the label (prognosis). Each vertical bar corresponds to one disease. The orange portion of the bar means the number of patients with this symptom while the blue portion means the number of patients without the symptom. Some of these features could be very revealing. For example, the "ulcers_on_tongue" almost deterministically indicates "GERD" in the data. Some other features do not have such differentiating power, for example, “high_fever” appears very

salient in many diseases and would not help as much to diagnose a disease individually. It has to be combined with other symptoms to indicate particular diseases more effectively. Similarly, "chest_pain" and "irritability" are also highly correlated to more than one "prognosis" and thus are not sufficient to narrow down to one disease. They can help reduce the match space quickly, though, so that other features can help further match the whole entry to a diagnosis.

Machine Learning Algorithms

Train and Test Data Split

We split the training data set into two parts: training data and test data. And we use the small input validation data to help evaluate the model's performance.

For one study, we do 70%-30% split. We can adjust the split ratio, and all the algorithms work on the same training and test data.

In our investigation, we picked 7 common ML algorithms as they are more widely used.

Algorithms

1. Decision Tree

This algorithm is robust against missing data, runs very fast and usually achieves high accuracy without much tuning. Its explainability is also very good.

2. Random Forest

This algorithm builds multiple decision trees to avoid local optimization issues. Itt runs slower than a single decision tree.

3. Gradient Boosted Tree

The gradient boosted tree algorithm can reduce the chance of bias compared to the Decision Tree method. It does run much slower since it iterates over many trees and builds things incrementally.

4. SVM

Support Vector Machine (SVM) is relatively light weight and could perform very well in high dimensional space.

5. Naive Bayes

Naive Bayes has an assumption that all the input features are independent of each other. Sometimes this assumption may not hold on real data.

6. Neural Network

Neural Network can predict any given function with reasonable approximation. For performance reasons, we used a shallow neural network with two hidden layers. The first layer contains 15 nodes (neurons) and the second layer contains 7 nodes.

7. Logistic Regression

It is very fast to train and not likely to overfit, although its performance depends on the number of features and size of input. It is widely used as the default algorithm with no prior knowledge.

Model Comparisons

We have a relatively small training, testing and validation data set here. And we will begin our comparison of the 7 models in terms of their accuracy and model training time. Note that here is the row size of each part of data:

Table 1. The training, test, and validation set size

Dataset	Row Size
Training	10332
Testing	4428
Validation	2952

Accuracy Comparison: The data set is not very big, but we can still hopefully find out how the models fare on their accuracies. And here are the model accuracies on test data and validation data:

Table 2. The accuracy of each algorithm based on test data and validation data

Algorithm	Accuracy on Test	Accuracy on Validation
Decision Tree	77%	88%
Random Forest	94%	95%
Gradient Boosted Tree	84%	82%
SVM	94%	95%
Naive Bayes	91%	93%
Neural Network	82%	87%
Logistic Regression	92%	93%

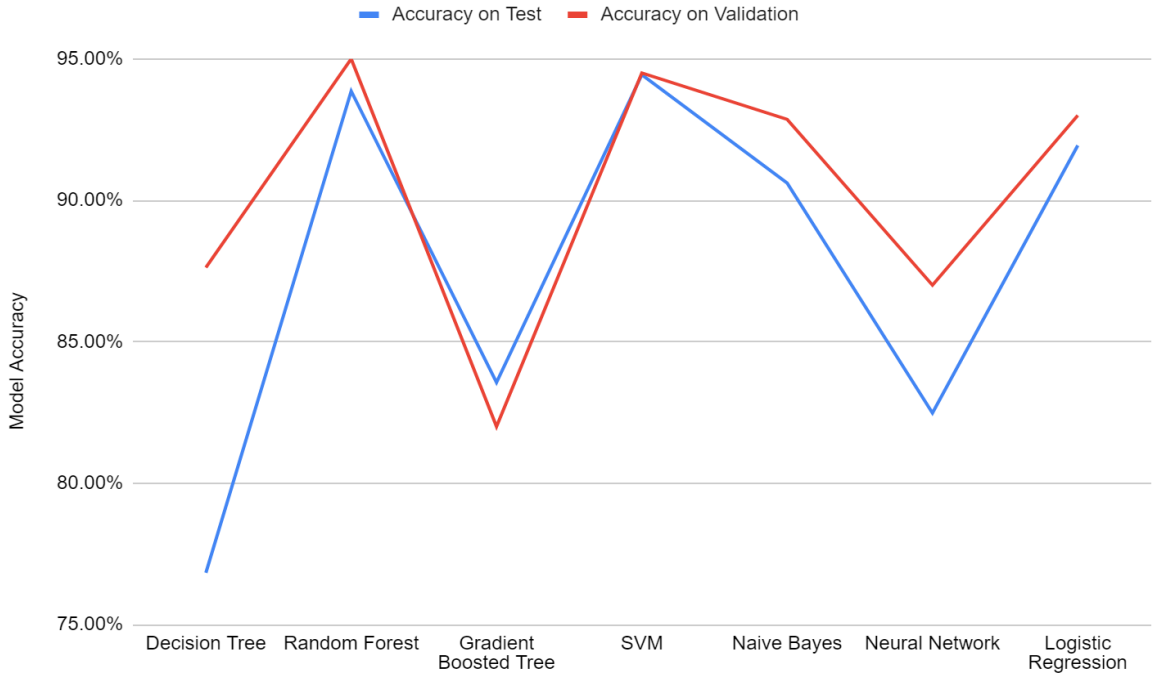

Figure 4. Visual plot of test and validation set accuracy of the 7 algorithms

We can see that although the data is pretty noisy, most models are pretty accurate on both the testing set and validation set. The Random Forest, SVM and Logistic Regression algorithms are doing well in learning the models, while the Decision Tree, the Gradient Boosted Tree and Neural Network algorithms took some hits. Compared to the simple Decision Tree, the Random Forest is doing better as it builds many trees than just one. And it is expected that the Neural Network would suffer from the noise in the data more than the other algorithms.

Confusion Matrix and RoC Curve: As we examined the dataset earlier, while the class samples are evenly distributed, it is very "unbalanced" for each individual class. We see nice accuracies of the models, but to ensure the models are not being biased towards any single class, we also checked the confusion matrix and RoC curves of the classifier results.

Confusion Matrix: Assume the label classes are in an array Ci, i = 0 .. 41, The rows are for the test data with the actual labels Ci, and the columns are the predicted label Cj. Each element e_{ij} in the matrix indicates the number of entries with actual label Ci but predicted label Cj.

One-vs-Rest (OvR) Metric: leverages a binary classification metric for multi-class classifications. For a given class, consider only the labels and predictions of this class as 1 and the other class as 0. And compute the corresponding metric as if this is a binary classification.

ROC Curve: It is a plot between the True Positive Rate (TPR) and False Positive Rate (FPR). And the area under the ROC curve is usually a robust indicator of the model's performance. Here we compute the TPR and FPR of each class using the One-vs-Rest (OvR) metrics and then take the average across the classes.

Here is an example confusion matrix on Neural Network model:

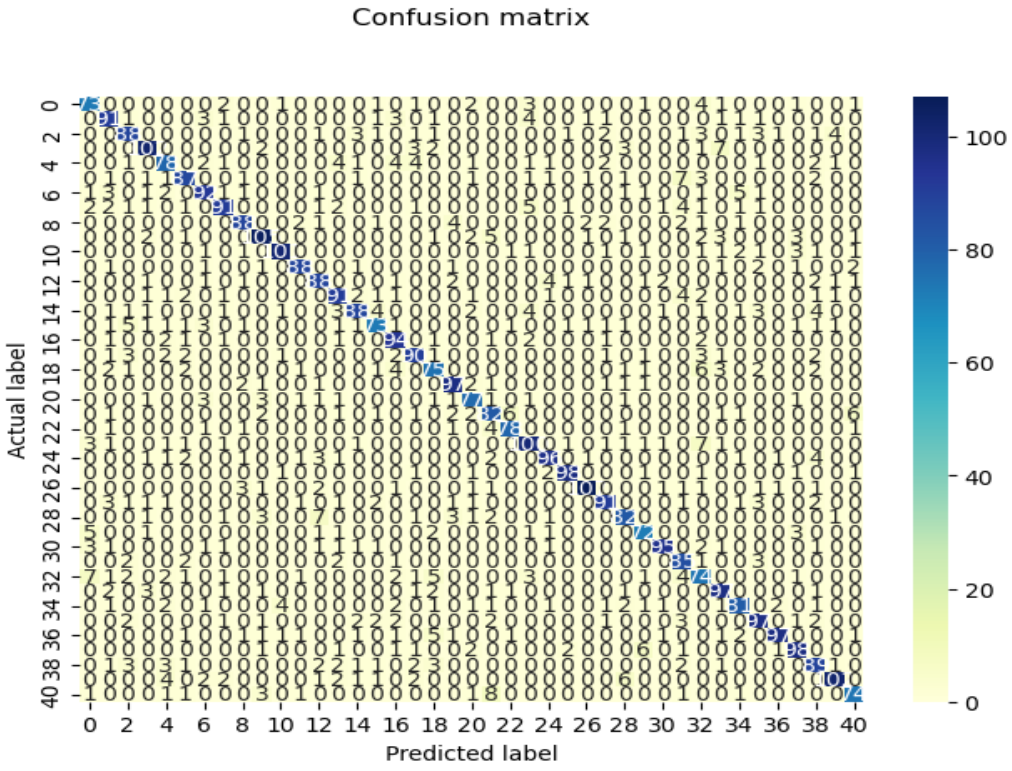

Figure 5. Confusion matrix of Neural Network algorithm

We can see quite some wrong predictions while the majority of the predictions are correct. This is expected since some of the features are not independent of each other, but of high correlation, and the training data contains quite some noise.

In order to be sure about the performance of the model, we also plotted the RoC curve of the model as below:

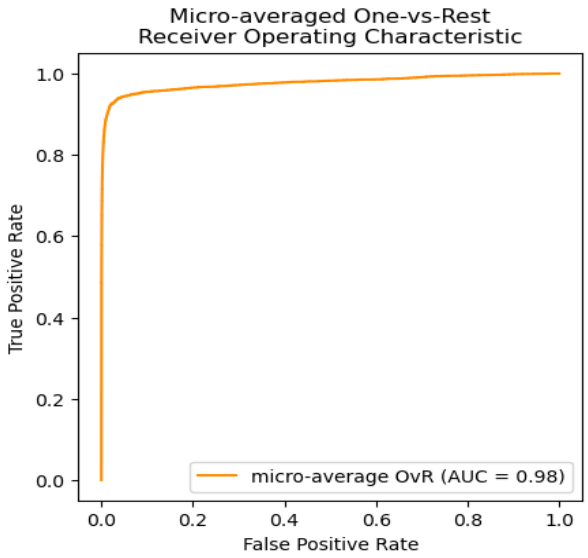

Figure 6. ROC curve of Neural Network model by averaged One-vs-Rest

We can see here the Neural Network model shows a great ROC curve. This is consistent with what we see in the confusion matrix.

We notice similar confusion matrices and RoC curves on other algorithms. For example, for Naive Bayes model,

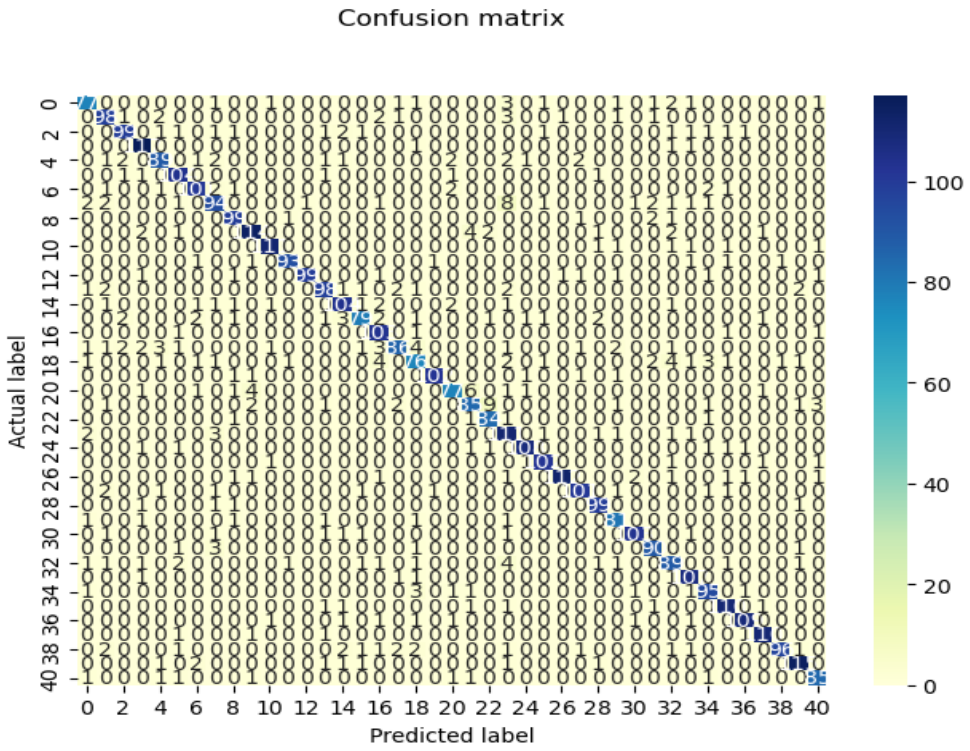

Figure 7. Confusion matrix of Naive Bayes Model

And ROC curve:

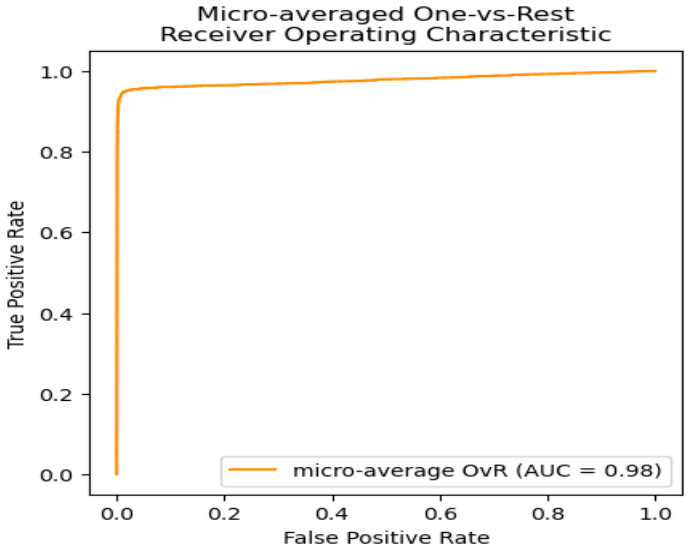

Figure 8. ROC curve of Naive Bayes model

This indicates that the models are actually “accurate”, with no systematic bias towards single classes.

Runtime Comparison: Due to the nature of using different algorithms, the model training time differs quite a bit.

Table 3. Model training time and test time

Algorithm	Model Training Time (ms)	Model Application Time (ms)
Decision Tree	167	4.84
Random Forest	2150	218
Gradient Boosted Tree	140000	357
SVM	40600	8820
Naive Bayes	62.2	155
Neural Network	89000	21.3
Logistic Regression	4460	11.8

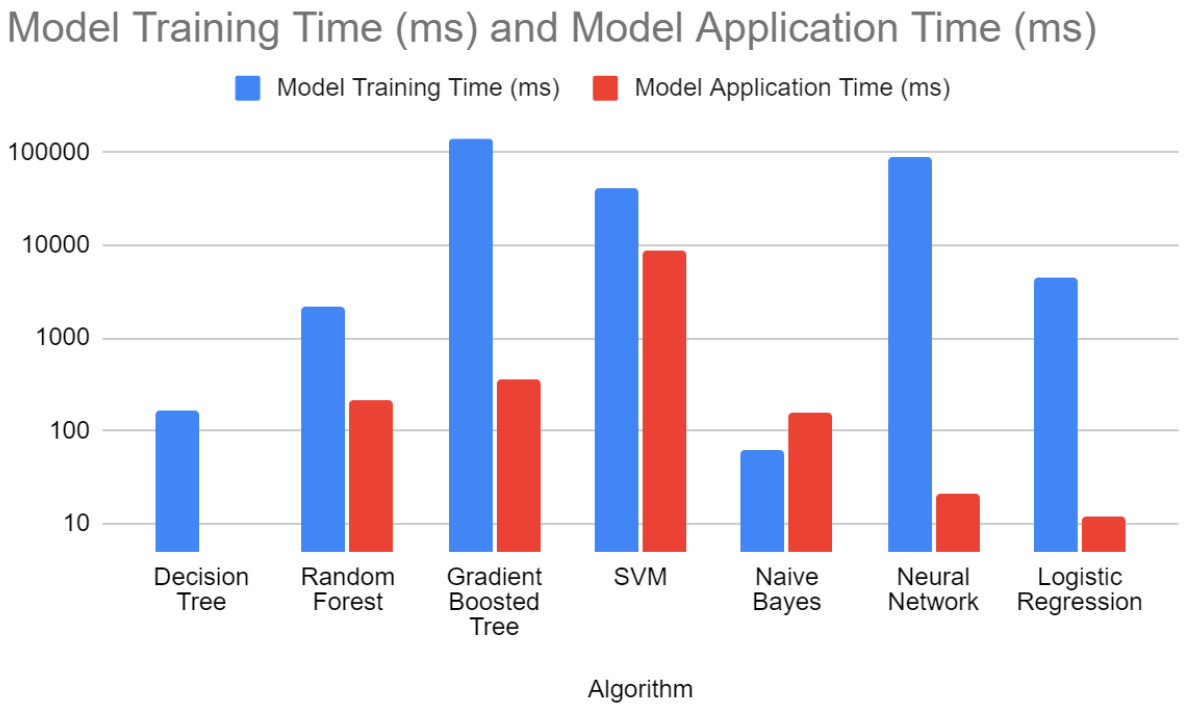

Figure 9. Bar chart of model training and application time (ms)

The values in the chart are at log scale. We can see that both Naive Bayes and Decision Tree models take very little time to train. The Random Forest, SVM, and Logistic Regression models take more time than the simple tree and Naive Bayes but still train a model in reasonable time. In comparison, the Gradient Boosted Tree and Neural Network (just 2 layers) are taking significantly more time to train a model.

When it comes to applying the models on the test set, the time is not proportional to the training time. This largely depends on the nature of the model and how much computation is needed to transform the test data. For example, SVM is very fast to train but the amount of computation during model application is quite big and causes the time to be significantly larger compared to, say, Logistic Regression.

If we take both times into account, Decision Tree is still most efficient, Logistic Regression and Naive Bayes are also very resilient on the time consumption.

Here is how the models perform with the accuracies and training time combined in one chart:

Table 4. Model accuracies and model training/testing time

Algorithm	Accuracy on Test	Accuracy on Validation	Model Training Time (ms)	Model Application Time (ms)
Decision Tree	77%	88%	167	4.84
Random Forest	94%	95%	2150	218
Gradient Boosted Tree	84%	82%	140000	357
SVM	94%	95%	40600	8820
Naive Bayes	91%	93%	62.2	155
Neural Network	82%	87%	89000	21.3
Logistic Regression	92%	93%	4460	11.8

We can see better on the model training time if we plot the training time in log scale:

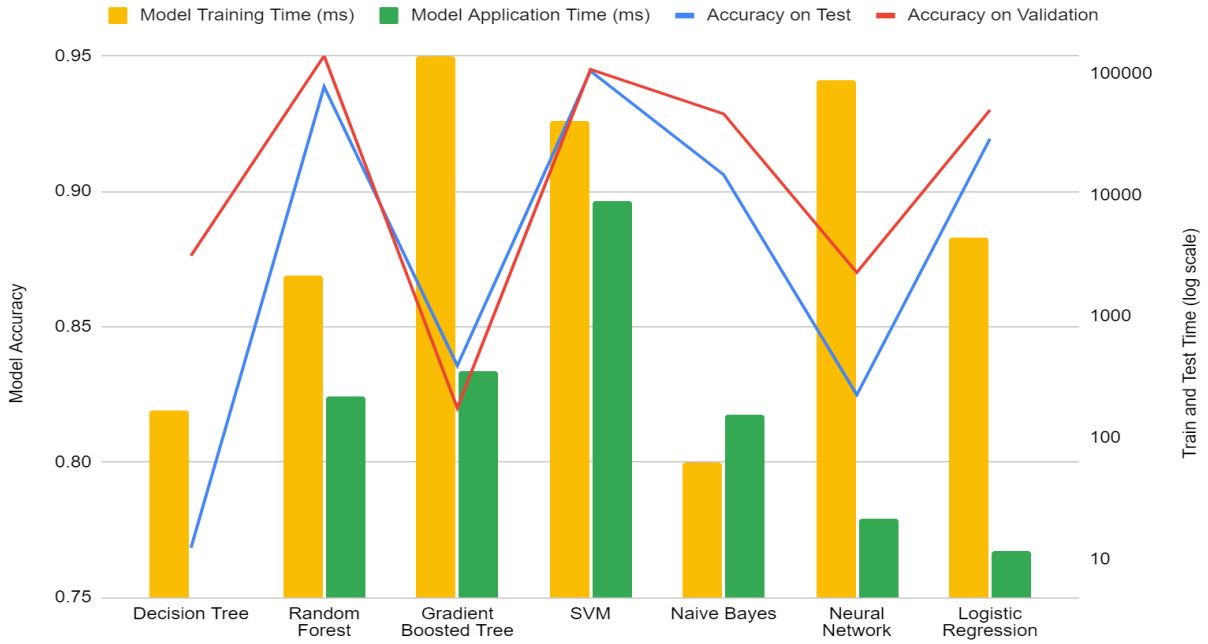

Figure 10. Model runtime and accuracy visual comparison

Other Considerations

We did our investigation on a relatively friendlier data set with nice numerical features. In reality, we often encounter data with missing data, invalid data, or skewed distributions. We usually need to do more data cleaning and normalization before we can apply some of the algorithms. In that regard, the tree based algorithms are superior since they are more robust in dealing with missing data, etc. They usually also converge very quickly with simple default settings. The more computation heavy methods like Neural Network have stricter requirements on input feature quality and need to be used with careful data examination.

Conclusion

In this investigation, we applied 7 different machine learning models on a disease classification task. We put no restrictions on the training process, so the methodology and simulations here are applicable to the more generic disease prediction cases. Our focus was on the multiclass unbalanced scenarios. We try to evaluate the methods based on their accuracies, model training time and model application time. Based on the investigations in this experiment, all the models are pretty accurate. In terms of the time it takes to train and apply the model, the simpler methods like Naive Bayes and Decision Tree are much more efficient to run. Overall, we would recommend starting with Decision Tree,Random Forest, and Logistic Regression models for disease prediction tasks. For more complicated data requiring more features, we can consider the more sophisticated models like Gradient Boosted Tree and Neural Network.

Acknowledgments

I would like to thank my advisor for the valuable insight provided to me on this topic.

References

1. Ahsan, M.M.; Luna, S.A.; Siddique, Z. (2022) Machine-Learning-Based Disease Diagnosis: A Comprehensive Review. In Healthcare (Basel). 10(3): 541 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/
2. Rett Syndrome https://www.mayoclinic.org/diseases-conditions/rett-syndrome/symptoms-causes/syc-20377227
3. Sklearn https://scikit-learn.org/stable/
4. Original data set https://www.kaggle.com/datasets/kaushil268/disease-prediction-using-machine-learning
5. Bemando, C., Miranda, E., Aryuni, M. (2021) Machine-Learning-Based Prediction Models of Coronary Heart Disease Using Naïve Bayes and Random Forest Algorithms. Proceedings of the 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), Pekan, Malaysia. pp. 232–237.
6. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A., San Tan, R. (2017) A deep convolutional neural network model to classify heartbeats. Comput. Biol. Med. 89, 389–396.
7. Charleonnan, A., Fufaung, T., Niyomwong, T., Chokchueypattanakit, W., Suwannawach, S., Ninchawee, N. (2016) Predictive analytics for chronic kidney disease using machine learning

techniques. Proceedings of the 2016 Management and Innovation Technology International Conference, Bang-Saen, Chonburi, Thailand. pp. MIT-80–MIT-83.

8. Asri, H., Mousannif, H., Al Moatassime, H., Noel, T. (2016) Using machine learning algorithms for breast cancer risk prediction and diagnosis. Procedia Comput. Sci., 83, 1064–1069.
9. Naz, H., Ahuja, S. (2020) Deep learning approach for diabetes prediction using PIMA Indian dataset. J. Diabetes Metab. Disord. 2020, 19, 391–403.
10. Neelaveni, J., Devasana, M.G. (2020) Alzheimer disease prediction using machine learning algorithms. Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India. IEEE: Manhattan, NY, USA. pp. 101–104
11. Chen, J., Wu, L., Zhang, J., Zhang, L., Gong, D., Zhao, Y., Chen, Q., Huang, S., Yang, M., Yang, X., et al. (2020) Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography. Sci. Rep. 2020, 10, pp 1–11.
12. Khan, M.A., Ashraf, I., Alhaisoni, M., Damaševi˘cius, R., Scherer, R., Rehman, A., Bukhari, S.A.C. (2020) Multimodal brain tumor classification using deep learning and robust feature selection: A machine learning application for radiologists. Diagnostics 2020, 10, 565