CSE 575: Statistical Machine Learning - Group 5

Arizona State University, Tempe, AZ

Project Title: Disease Prediction with Symptoms

Team Members:

1. Pratik Giri (ASU ID: 1224255318)
2. Nikitha Vipala (ASU ID: 1222193674)
3. Sarika Naidu Chiriki (ASU ID: 1224450630)
4. Sreya Sukhavasi (ASU ID: 1224567162)
5. Gowthami Radha Ananthaneni (ASU ID: 1224235025)

Roles:

1. Pratik Giri - My role was in Literature review, preliminary dataset analysis. I contributed towards milestones such as feature engineering for our dataset, modeling, building and testing of Neural Network models and comparative analysis of different models.
2. Nikitha Vipala - My role was to work on the preliminary data analysis, writing module for K Nearest Neighbors algorithm, and also contributed to the comparison of the different algorithms.
3. Sarika Naidu Chiriki - My role was to work on data analysis, preprocessing the data, and testing it with data, and worked with the fellow team members on analyzing the results from different algorithms.
4. Sreya Sukhavasi: My role was to work on preliminary data analysis, modeling, building and testing Linear Regression algorithm, and compared comparative analysis of different models
5. Gowthami Radha Ananthaneni - My role was to work on preliminary data analysis, building and testing the Decision Tree classifier, and comparing results derived from different algorithms.

Introduction

Summary of the Problem

The healthcare sector has grown significantly in size. The healthcare sector generates enormous amounts of healthcare data every day, from which it is possible to extract knowledge for diagnosing diseases that can affect a patient in the future utilizing treatment information and health data. Later, this concealed information in the healthcare data will be used to make decisions that will improve the patient's health. In this project, we are using this data to predict the disease of a patient based on the symptoms experienced by them. The major challenge in prediction is to accurately represent the complex health data information so that the prediction will be feasible. The approach of this project is to use machine learning and a neural network to develop a predictor for this disease and get a prediction of the disease. To get accurate results, we implemented five techniques for the prediction. Out of all the algorithms, we can predict diseases accurately with the help of a neural network.

Previous Work:

1. *S. Grampurohit and C. Sagarnal, "Disease Prediction using Machine Learning Algorithms," 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-7, doi: 10.1109/INCET49848.2020.9154130*

   Methods and Results:

   In this research paper, the techniques of machine learning have been employed in assorted applications including Disease prediction with the aim of developing a classifier system using machine learning algorithms to immensely help to solve the health-related issues by assisting the physicians to predict and diagnose diseases at an early stage. A Sample data of 4920 patients' records diagnosed with 41 diseases was selected for analysis. A dependent variable was composed of 41 diseases. 95 of 132 independent variables(symptoms) closely related to diseases were selected and optimized. This research work carried out demonstrates the disease prediction system developed using Machine learning algorithms such as Decision Tree classifier, Random forest classifier, and Naïve Bayes classifier. This paper inspired us a lot to employ similar models as well as some different models to be used for Disease Prediction algorithms.

2. *Ramalingam, V. V., Ayantan Dandapath, and M. Karthik Raja. "Heart disease prediction using machine learning techniques: a survey." International Journal of Engineering & Technology 7.2.8 (2018): 684-687*

   Methods and Results:

   This research paper focuses on building Machine Learning models for Heart Diseases or Cardiovascular Diseases (CVDs) prediction on the data collected by many medical organizations around the world. In this research paper, authors have made use of models like Naive Bayes, SVM, KNN, Decision Tree, etc.

Motivation

As humans are affected due to various diseases in the whole world. It is currently a great challenge as these diseases are causing a lot of deaths in recent years. Though there are models available for disease prediction, there is always a lack in the prediction models while considering the accuracy. The models can have more accuracy in predicting the diseases. WHO has recently declared that around million of population death is occuring due to several known and unknown diseases. So it will be a great achievement if we implement a model for high accuracy of disease prediction mainly similar to the COVID 19 in order to reduce the impact of such diseases. Inorder to treat the disease, identifying the disease is the first step. When testing resources are not available at the moment, and the patient is having a very serious case, this system helps the patient to get a clear picture on the situation and helps to take necessary steps to eradicate the problem. The main motivation and aim is to find a good prediction algorithm and compare them for the best accuracy in order to predict the disease.

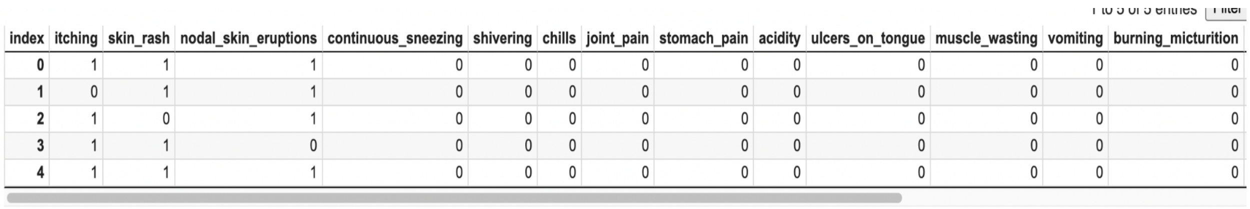
Problem description

In this fast-paced world, taking care of oneself is more complicated than we imagine. Most people do not realize how health is important unless it is too late. Identifying the problem is the first step to fighting it. In their busy lives, taking time to get their body checked once in a while is vital, but most people do not incorporate this step in their plans due to workload or external factors. Even recently there was a pandemic issue due to COVID all over the world is a global concern and many of the victims have not been provided with efficient treatment. Also, a medical specialist cannot always be available, and a doctor can often diagnose based on the medical examination findings done with these models. Therefore, a disease prediction system is often built to help patients at different stages in predicting disease.

With the evolution of Big Data technology in this era, predicting diseases has been given more attention as health comes before anything. Hence the main focus is on diagnosing and identifying diseases using Machine Learning and deep learning techniques. Hence, we have decided to use efficient algorithms, to help patients in using the applications to predict the disease before it is too late. As there are many techniques that can be incorporated with this problem, we will be comparing the results of the techniques we have chosen in terms of accuracy and performance and provide a good solution based on the situation. Furthermore, our model can predict the disease based on symptoms and help prevent a disease that would likely have worse impacts in the future.

Dataset:

We will be using the data set available at:
https://www.kaggle.com/kaushil268/disease-prediction-using-machine-learning. This dataset is divided into training and testing dataset. Each of these datasets has 133 columns. This dataset enables prediction of different types of diseases such as Diabetes, Hypertension, Heart Attack, etc. The training dataset is very balanced as each of the 41 possible diseases have equal 120 numbers of data points with a total of 4920 data points.



1 to 5 of 5 entries Filter

index	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	muscle_wasting	vomiting	burning_micturition
0	1	1	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	0	0	0	0	0	0	0	0	0	0

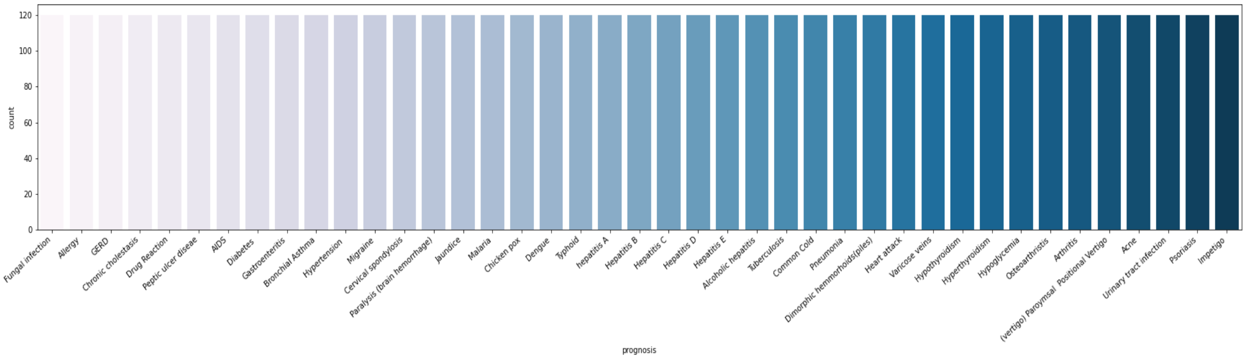


Figure 1: Dataset description

Data Preprocessing:

Data preprocessing is a technique used to clean up raw data. The aim of preprocessing is to identify the important data that fits into our models. Since the data is so large, we will be extracting the crucial data from the dataset chosen above. We will divide the data for training and testing, making sure that the data is compatible with all the algorithms that we are planning to implement.

Methodology

Linear Regression:

Linear Regression is a supervised machine learning technique. It is used to predict a dependent variable(Y) based on an independent variable(X). Here, the symptoms are the independent variable and the probability of having the disease is the dependent variable. This regression method is used to find out the linear relationship between 'X' and 'Y' [4].

In our dataset the 'Y' variable is 'Prognosis', which is not a numerical value. Linear regression is used if we want to predict continuous values or quantitative responses[3]. There are 41 types of diseases in the prognosis column. To indulge our dataset 'Y' variable into a linear regression model we have leveraged one-hot encoding for the 'Prognosis' column. That is, it will create 41 more columns, each column representing a particular disease with binary value in the data(1- have that disease, 0 - doesn't have the disease).

Results:

The model performance on the test data is 87.41% where on the training data it is 97.41%. The Mean Absolute Error of the test data is 1.19 and is 1.17 in the training data. Mean Squared Error of the test data is 18.38 and is 3.62 in the training data. Root Mean Square Error of the test data is 4.29 and is 1.90 in the training data. The figure 2 below shows us the plot of Y predicted against the Y test.

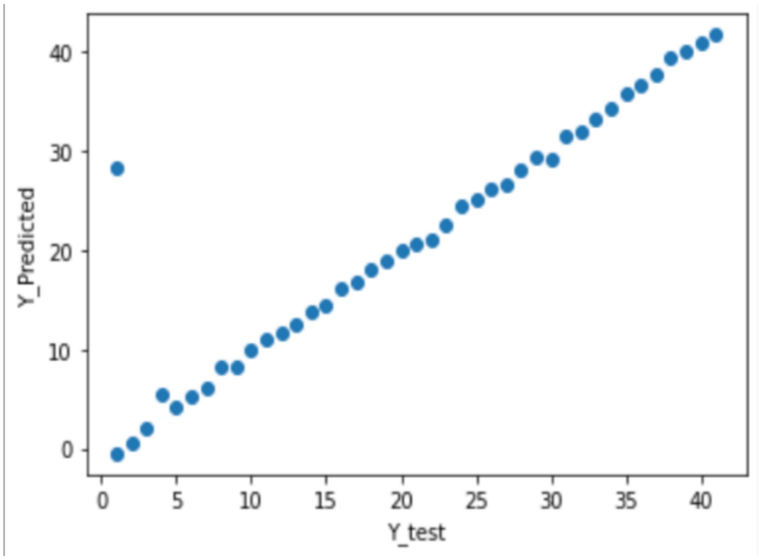


Figure 2: Y_predicted vs Y_test in Linear regression

Decision Tree:

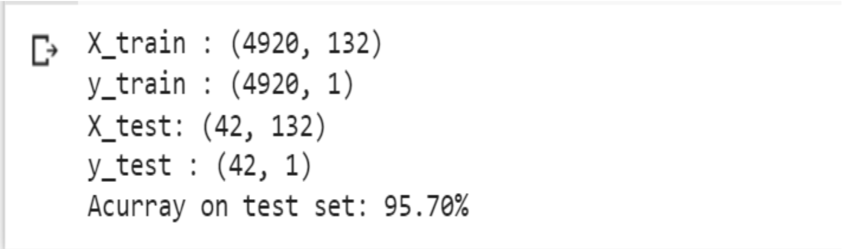
Decision trees are a powerful data mining method that many data mining scientists have used. They are an effective statistical technique for analysis and learning that combines multiple explanatory parameters to predict an outcome. Decision trees are highly complex because they must deal with multiple explanatory variables to create a prediction model. They deal with numeric, categorical, or ordinal variables, and they are useful for multiple types of data. Decision trees effectively predict the outcome from a large number of heterogeneous factors. It takes the data, places it in the root node, and then looks at it from multiple angles. It will then split the input based on true/false questions and find the correct outputs based on the inputs. So, the algorithm first learns the input and then comes up with the output. The decision trees algorithm is very effective in finding the best decision rules[5]. The decision tree algorithm can find multiple decision rules, which is difficult with other algorithms.

A decision tree takes symptoms and predicts a disease based on that. This prediction is a classification of a specific disease. To get all the diseases for a particular symptom, we can pass them all to the Model, and the highest probability from all the diseases will be the final decision[6]. The Model for predicting a disease will take input parameters and then predict the disease. The Model is responsible for the decision-making process. After that, we get the output from the Model and then return the predictions to the user. In this project, we developed a model that uses the Gini Impurity to split the dataset into a series of decisions. Gini Impurity measures how well a model can separate items into two categories. When learning from labeled data, Gini

Impurity is ideal because it will decrease as models start to learn the correct assignments. However, Gini Impurity can be high if data were labeled based on random guesses or if the Model has a large variance in assigning labels.

Results:

The Decision tree works with the underlying symptoms and predicts a disease. The symptoms are passed as input to the model for predicting the disease.This array matches the disease data collection and ends at a common leaf node with the highest degree of trust. With the help of decision trees, we are able to obtain the accuracy of 95.7%.



```
X_train : (4920, 132)
y_train : (4920, 1)
X_test: (42, 132)
y_test : (42, 1)
Acurray on test set: 95.70%
```

Figure 3: Accuracy of decision tree

As the tree is very big because of multiple input attributes, Attaching the start of a tree only.

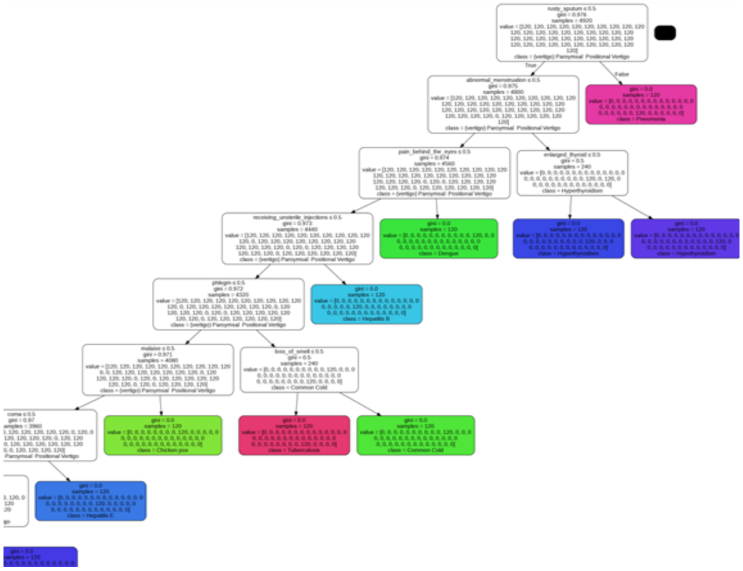


Figure 4: Decision tree with input attributes

K-nearest neighbor:

K-Nearest Neighbor is one of the supervised algorithms where it relies on the labeled data and is used for classification problems. This algorithm is generally used for different label types, data sizes, ranges and noise levels. And also as the data is based on the real world scenarios for predicting the disease using the symptoms. KNN [2] is used in many applications such as problem solving, interpretation and classification, function learning, and teaching. We have a set of features and we also have the output which is associated with the given features. We use this data and try to predict the output using the features associated with it along with the data. Here K represents the variable defining the number of nearest neighbors to be considered for classification. This algorithm is simple as it classifies the new input features into corresponding output labels based on the similarity between the k neighbors. As it stores all the cases from previous predictions and some of the training data it will be easy to calculate the distance between the features and predict the output. It runs a query to find the similarities from the training set and checks the neighbors closest to the query in the testing. Once the query is done, the algorithm runs a voting rule for the majority checking and assigns the label accordingly.

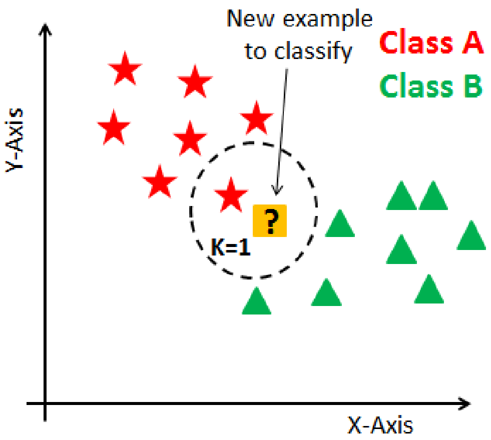
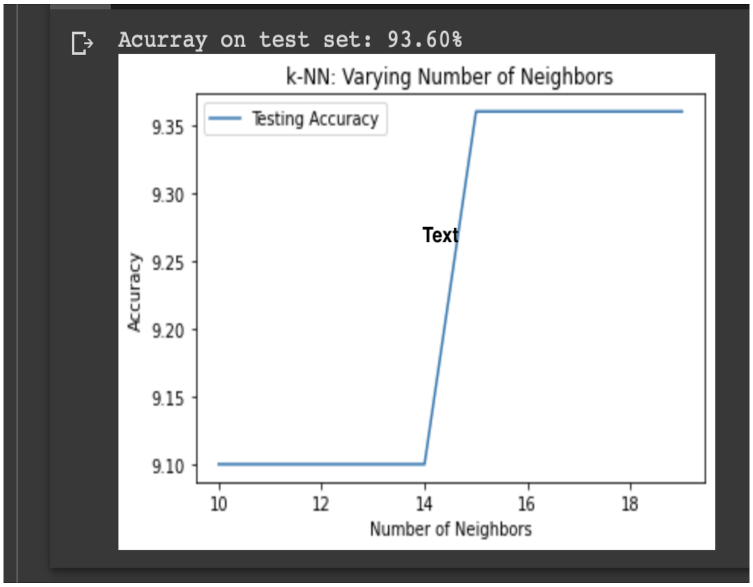


Figure 5: K-Nearest Neighbors [1]

In this algorithm we follow the following steps for implementation. Initially we have chosen a k value to be 5 and then divided the training and testing data set. Removed the output label from the test data set and picked the euclidean distance as the distance measure. Once the data was preprocessed then we find the euclidean distance between the k neighbors. The check for the nearest neighbor with the similarities in the features using the distance measure. Once the K neighbors are found within the training data, we already know the label associated with the training set. We then use this labeled data to run over the voting to check which class has the majority votes based on the similarities. If there are ⅗ neighbors which predicts a particular diseases for instance COVID19 based on the features similarities and distance measure, we predict the output label of the testing data to be COVID 19 disease. We then find the accuracy for each testing data and store it. We now increase the K value from 5 till 20 and then stop the algorithm as the accuracy has leveled off at K value 20.

Results

As there was an increase in the K value there was an increase in the accuracy. We have stopped the K value increment at 20 as the accuracy value leveled off after that. We have stored the accuracy values in the list for each K value and then plotted a graph for K vs accuracy. Here size and features were also important factors for the increased accuracy in the KNN. This algorithm predicted the diseases with an accuracy of 93.6%.



```
Acurray on test set: 93.60%
```

Figure 6: Accuracy vs number of neighbors graph in KNN

Neural Network

1. **Multilayer Perceptrons (MLP)**

   The field of artificial neural networks is often just called neural networks or multi-layer perceptrons after perhaps the most useful type of neural network. A perceptron is a single neuron model that was a precursor to larger neural networks. It is a field that investigates how simple models of biological brains can be used to solve difficult computational tasks like the predictive modeling tasks we see in machine learning. The goal is not to create realistic models of the brain but instead to develop robust algorithms and data structures that we can use to model difficult problems.

   Multi layer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers—the input layer, output layer and hidden layer, as shown in Fig. 7. The input layer receives the input signal to be processed. The required task such as prediction and classification is performed by the output layer. An arbitrary number of hidden layers that are placed in between the input and output layer are the true computational engine of the MLP. Similar to a feed forward network in a MLP the data flows in the forward direction from input to output layer. The neurons in the MLP are trained with the back propagation learning algorithm. MLPs are designed to approximate any continuous function and can solve problems which are not linearly separable. The major use cases of MLP are pattern classification, recognition, prediction and approximation.

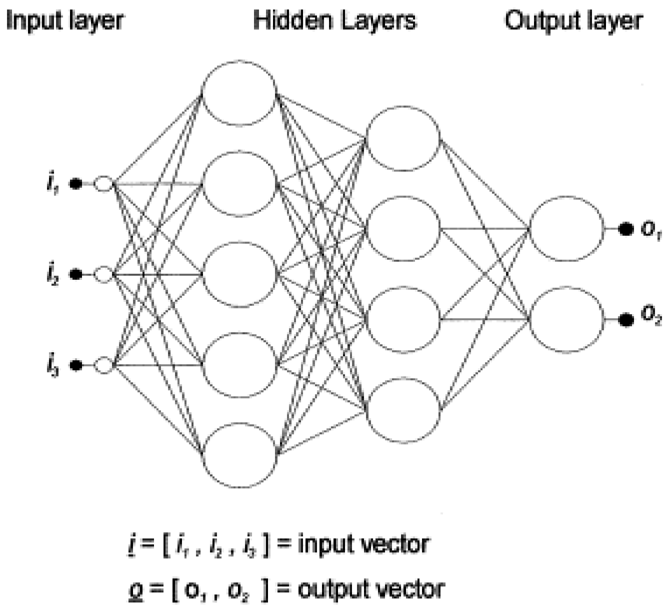
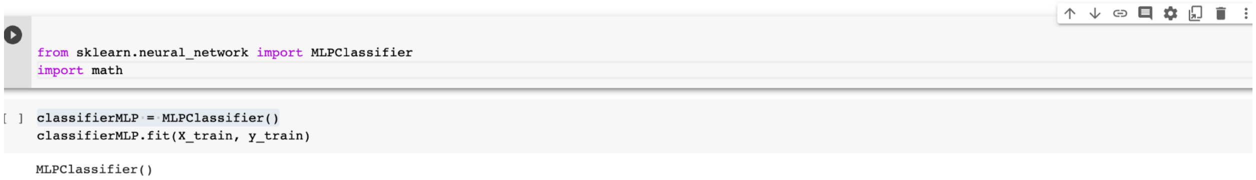

Input layer
Hidden Layers
Output layer
i_1
i_2
i_3
o_1
o_2
$\underline{i} = [\, i_1 , i_2 , i_3 \,]$ = input vector
$\underline{o} = [\, o_1 , o_2 \,]$ = output vector

Figure 7: Schematic representation of a MLP [7].

In our implementation of MLP, we have used scikit-learn’s MLPClassifier() method with all the default parameter values such as hidden_layer_sizes, activation (default = ‘relu’), solver (default=’adam’), etc.

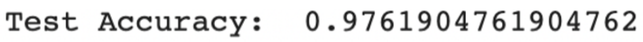


```
from sklearn.neural_network import MLPClassifier
import math
```

```
[ ] classifierMLP = MLPClassifier()
    classifierMLP.fit(X_train, y_train)

    MLPClassifier()
```

Figure 8: Code implementation of Scikit-Learn’s MLPClassifier()



```
Test Accuracy:  0.9761904761904762
```

With this implementation, we achieved the test accuracy of **97.6%**.

2. Deep Learning

Lastly, we have made use of Deep Learning. Most modern deep learning models are based on artificial neural networks. Deep learning is a branch of machine learning (ML) that mimics the functioning of the human brain to find correlations and patterns by processing data with a specified logical structure. Also referred to as deep neural networks, deep learning uses multiple hidden layers in the neural network as opposed to traditional neural networks that only contain a handful of hidden layers. The word “deep” in deep learning refers to the number of layers.

Deep learning has advantages over traditional models like elimination of need of feature engineering and data labeling. While training deep learning models can be cost-intensive, once trained, it can help businesses cut down on unnecessary expenditure. Also, deep learning is highly scalable due to its ability to process massive amounts of data and perform a lot of computations in a cost- and time-effective manner. This directly impacts productivity (faster deployment/rollouts) and modularity and portability (trained models can be used across a range of problems).

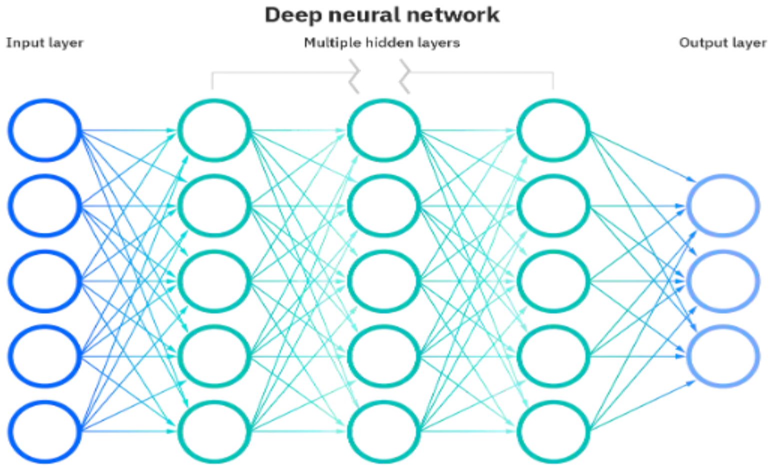


Figure 9: Schematic representation of Deep Learning Network. [8]

In our implementation of Deep Learning, we have used *ReLU* as the activation function and *adam* as optimizer. The learning is done in batches with batch size as 30. Below Figure 10 shows the relationship between the number of epochs and Model accuracy. We can observe that the accuracy keeps increasing as the number of epochs increases.

By following this model with given parameters, we were able to achieve the highest accuracy of 99.1% as shown in Figure 11.

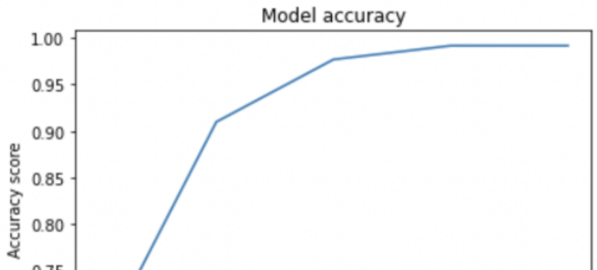
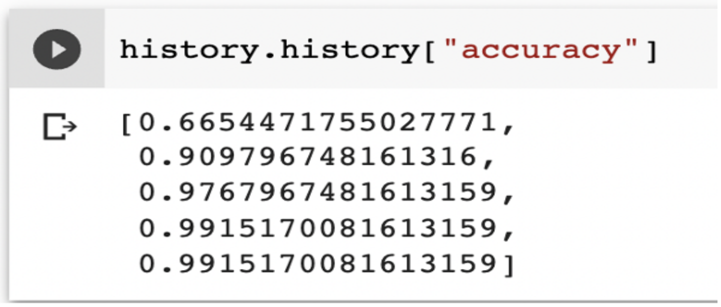


Figure 10: Graph of Model Accuracy vs. Number of Epochs



```
history.history["accuracy"]
```

```
[0.6654471755027771,
 0.909796748161316,
 0.9767967481613159,
 0.9915170081613159,
 0.9915170081613159]
```

Figure 11: Accuracy over different epochs

Comparative Analysis

- KNN was chosen as it has an advantage that it does not require any correlation between target variables and features. After running the model and analysis the accuracy has increased by almost 5%.
- We choose a decision tree as missing values in the data do not affect the process of building a decision tree to any considerable extent. We got an accuracy of 95.7%.
- Although the decision tree model gave better accuracy compared to KNN and linear regression, we decided to develop a Neural Network model as well, as a small change in the data can cause a large change in the decision tree structure, causing instability.
- The Deep Learning model was chosen as it works best when the relation between the input and output is non-linear, which is the case for our dataset.

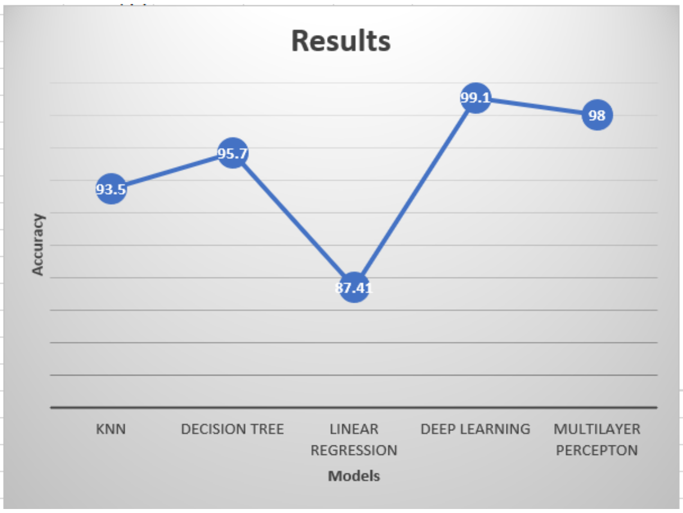


Figure 12: Comparison of Models for disease prediction

- Among all the models implemented, Deep Learning model predicted the diseases with highest accuracy of 99.1%

Conclusion

The project presented a technique to predict the disease based on symptoms of a patient successfully. The main contribution is to compare different models for predicting the disease and find the best model based on the accuracy of predicting at an early stage. The actual accuracy was also increased due to the data pre processing methods used in order to eliminate the empty values and handle corrupted values. This model will help in reducing the cost and improving the recovery process of any patient. The Neural network model gave the highest accuracy of 99.1% using the factors mentioned. For the other models the accuracy was low due to the dependency on the parameters as well as comparatively less data points. In the Deep Learning model, successfully overcame overfitting by trying different sets of hidden layer dimensions.

Future work

The models used a dataset which is comparatively smaller. The accuracies of these models, especially Neural Network, can be improved with larger dataset. Any Artificial Intelligence technique can be used such as Prophetic analytics to predict diseases more efficiently. Some more models can be implemented with the larger dataset such as Random Forest, RNN, LSTM, etc. The future scope can also be further increased by using fuzzy knowledge in order to find the exact patterns.

References

[1] Rajvi Shah, “Introduction to k-Nearest Neighbors (kNN) Algorithm”, Mar 3, 2021.

[2] Pan, Z., Wang, Y. & Pan, Y. A new locally adaptive k-nearest neighbor algorithm based on discrimination class. *Knowl. Based Syst.* 204, 106185 (2020).

[3] M. Huang, "Theory and Implementation of linear regression," 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), 2020, pp. 210-217, doi: 10.1109/CVIDL51233.2020.00-99.

[4] Xin Yan and Xiaogang. Su, Linear regression analysis: theory and computing, World Scientific, 2009.

[5] Kondababu, A., Siddhartha, V., Kumar, B. B., & Penumutchi, B. (2021). A comparative study on machine learning based heart disease prediction. *Materials Today: Proceedings*.

[6] Krishnamoorthi, R., Joshi, S., Almarzouki, H. Z., Shukla, P. K., Rizwan, A., Kalpana, C., & Tiwari, B. (2022). A novel diabetes healthcare disease prediction framework using machine learning techniques. *Journal of Healthcare Engineering*, *2022*.

[7] M.W Gardner, S.R Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmospheric Environment, Volume 32, Issues 14–15, 1998, Pages 2627-2636, ISSN 1352-2310, https://doi.org/10.1016/S1352-2310(97)00447-0.

[8]IBM Cloud Education, http://www.ibm.com/cloud/learn/neural-networks, 17 August 2020