Disease Prediction Using Machine Learning

Anshu Yadav

January, 2021

# 

# 

[Overview](#_qtfz804nupyu)

[Problem Statement](#_hweaelp6kmy)

[Formulation](#_um1384ux3g5p)

[Analysis](#_qjwn9hswn33h)

[Data Wrangling](#_j5su424lb7fu)

[Exploratory Data Analysis](#_lv2zznp2ydh)

[Evaluation Metrics](#_nzv89ar9555j)

[Data Modeling](#_uildgju98g8e)

[k-Fold Cross-Validation](#_816r7gdn87za)

[Neural Network](#_uqchazcsv2ly)

[Conclusions](#_7wnkjrmlgfko)

[Future Research](#_njnju1na0goh)

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# Overview

Machine Learning is an emerging approach that helps in prediction, diagnosis of a disease. This paper depicts the prediction of disease based on symptoms using machine learning. Machine Learning algorithms such as Naive Bayes, Decision Tree and Random Forest are employed on the provided dataset and predict the disease.The source data for disease prediction dataset was chosen from kaggle.

# Problem Statement

This project is about prediction of disease based on symptoms using machine learning. Machine Learning algorithms such as Naive Bayes, Decision Tree and Random Forest are employed on the provided dataset and predict the disease. Its implementation is done through the python programming language. The research demonstrates the best algorithm based on their accuracy. The accuracy of an algorithm is determined by the performance on the given dataset.

## Formulation

The dataset, after pre-processing and multiplication, contains around 4920 rows with 131 unique symptoms. This was done to tackle the problem of only having a single row for each disease which results in poor training of data. This idea was inspired by the real-world scenario where a patient even showing some of the symptoms of all the symptoms for a disease can be suffering from that disease, therefore it is a logical extension of the dataset.

# Analysis

Final count of diseases in the dataset were a total of 41 and 131 symptoms. To multiply the dataset, each disease’s symptoms are picked up, combinations of the symptoms are created and added as new rows in the dataset.

## Data Wrangling

Following Information regarding the symptom and disease are provided:

1. Complete Dataset consists of 2 CSV files . One of them is training and the other is testing your model.
2. Each CSV file has 133 columns. 132 of these columns are symptoms that a person experiences and the last column is the prognosis.
3. These symptoms are mapped to 42 diseases you can classify these set of symptoms to.

The below list of cleanup tasks were performed on the data:

1. Dataset came into two sets (train and test set). Training dataset has 4920 rows and 134 columns whereas the test dataset has 42 rows and 133 columns. All the columns between train and test dataset were similar except the “**Unnamed: 133”** column which we deleted from the training dataset also. It has NaN values.
2. Data wrangling was done on a training data set.
3. None of the features contain null values.
4. Each disease has a positive sample size of 120 patients.
5. The overall size of the dataset is a bit low given that it only has ~4k so the machine learning component may be limited, especially given that there are 42
6. All the columns except prognosis have value either 0 or 1. 1 means symptoms related to a particular disease and 0 means not a symptom of that particular disease.

0 = no 1 = yes

## 

## Exploratory Data Analysis

The data dictionary file provided with the dataset, indicates that columns are information about the symptoms and the outcome of prognosis.

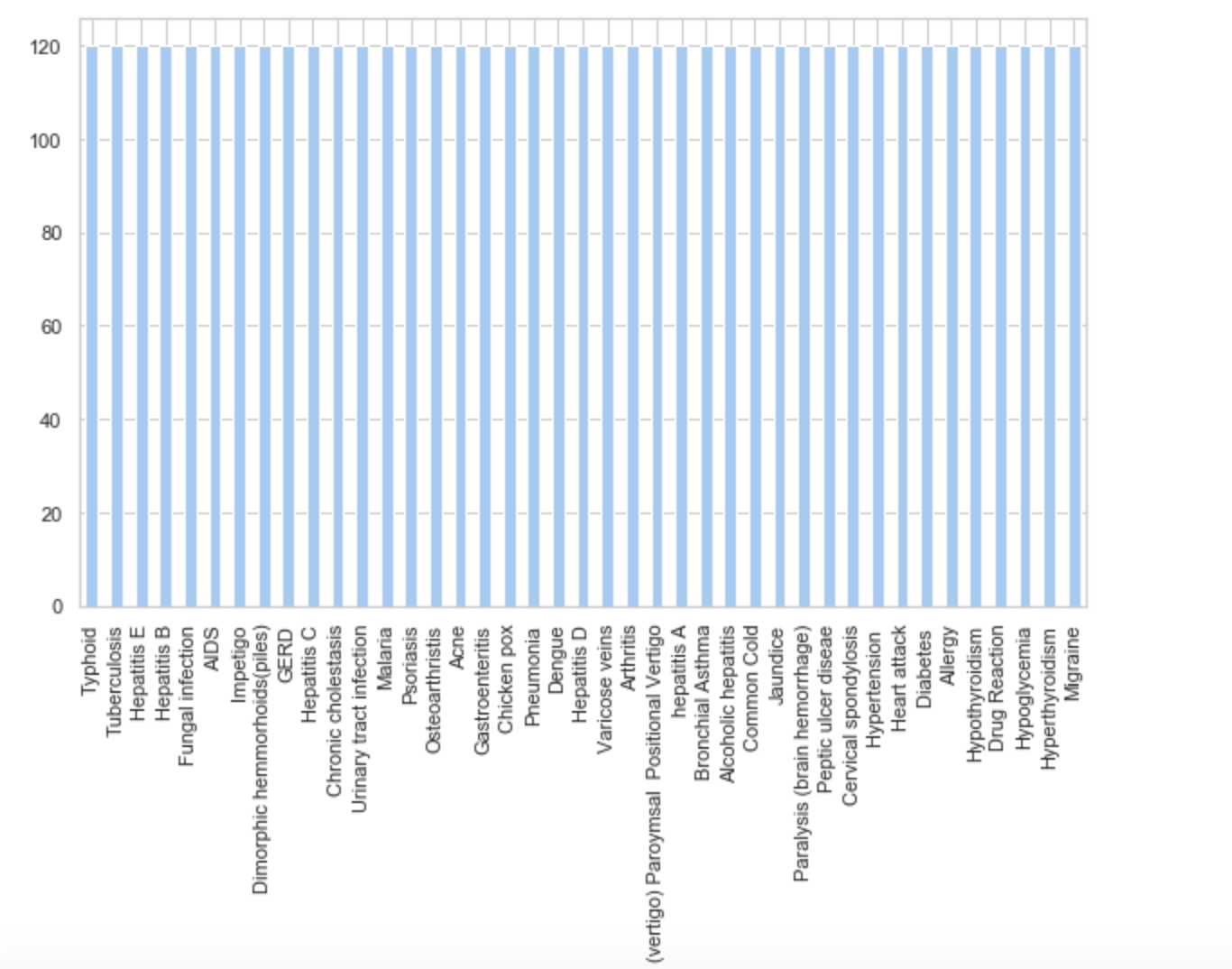
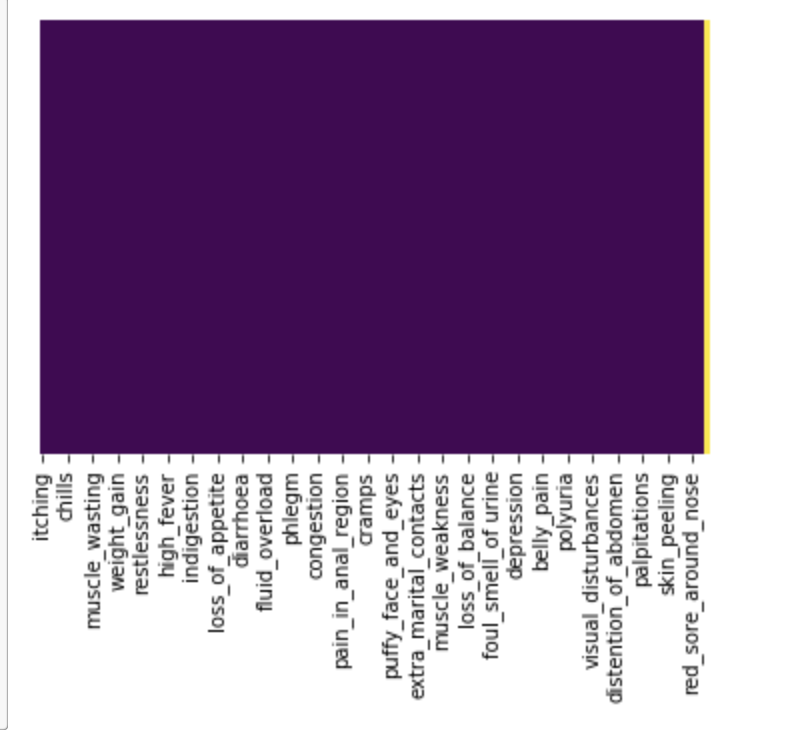


Figure 1: Representation of positive sample size on the basis of disease

A closer look at the data identifies some NA and “?” values that will need to be addressed in the cleaning step. We also want to know the number of observations in the dependent variable column to understand if the dataset is relatively balanced.

Figure 2: Heatmap of training dataset of 134 columns to see if null values.

Histogram of particular diseases and columns showing symptoms for that particular disease. I retrieve all records for given input disease and then remove columns which are not its symptoms.

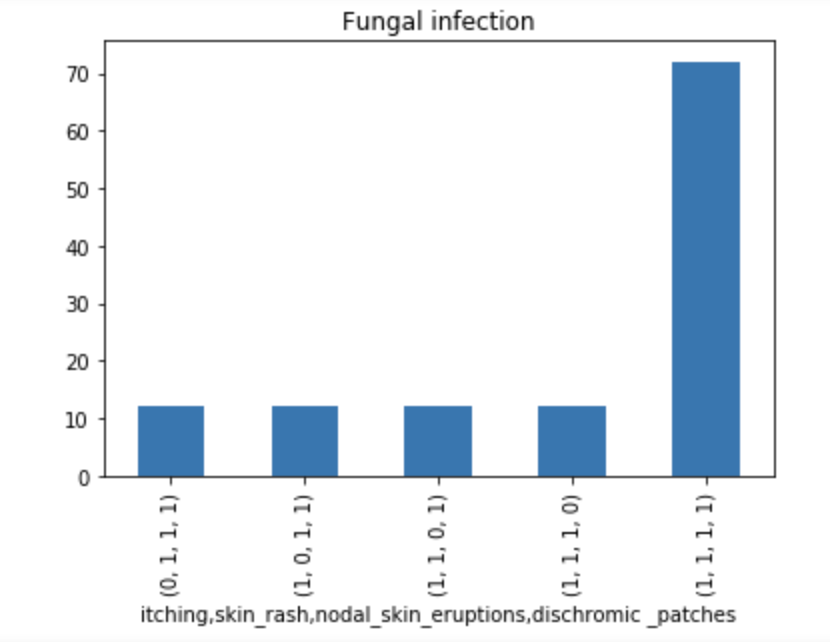


Figure 3: Histogram of all the symptoms of “Fungal Infection”

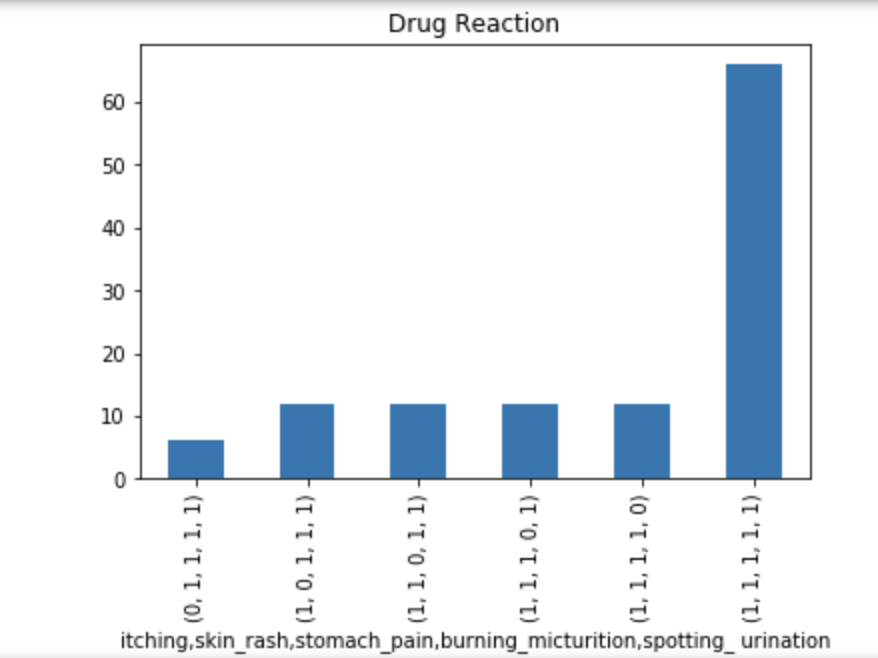


Figure 4: Histogram of all the symptoms of “Drug Reaction”

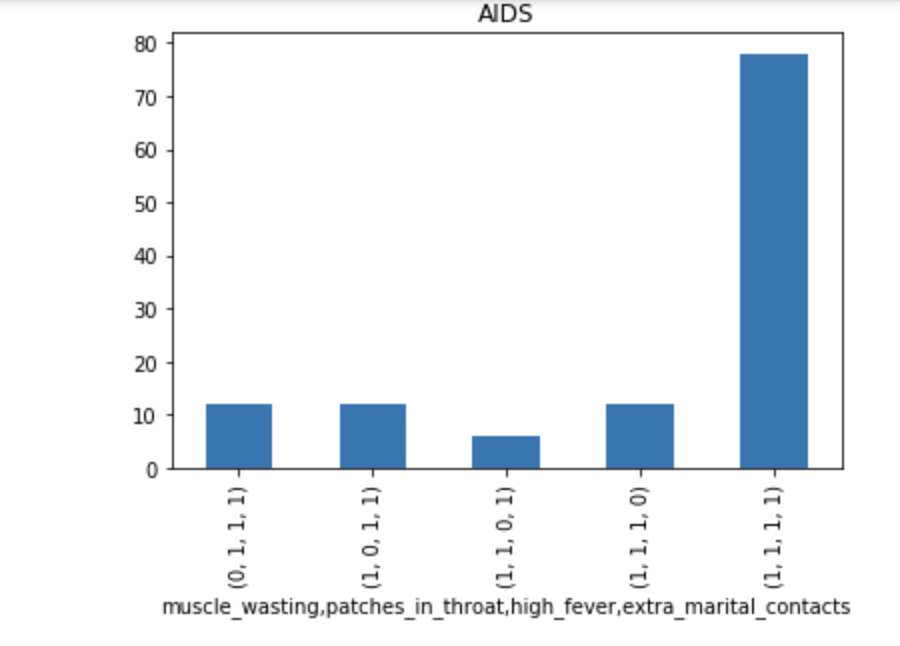


Figure 5: Histogram of all the symptoms of “AIDS”

Similarly all other 40 diseases were explored on the basis of positive symptoms to show specific disease.All bar plots of specific prognosis are attached in the appendix 1.

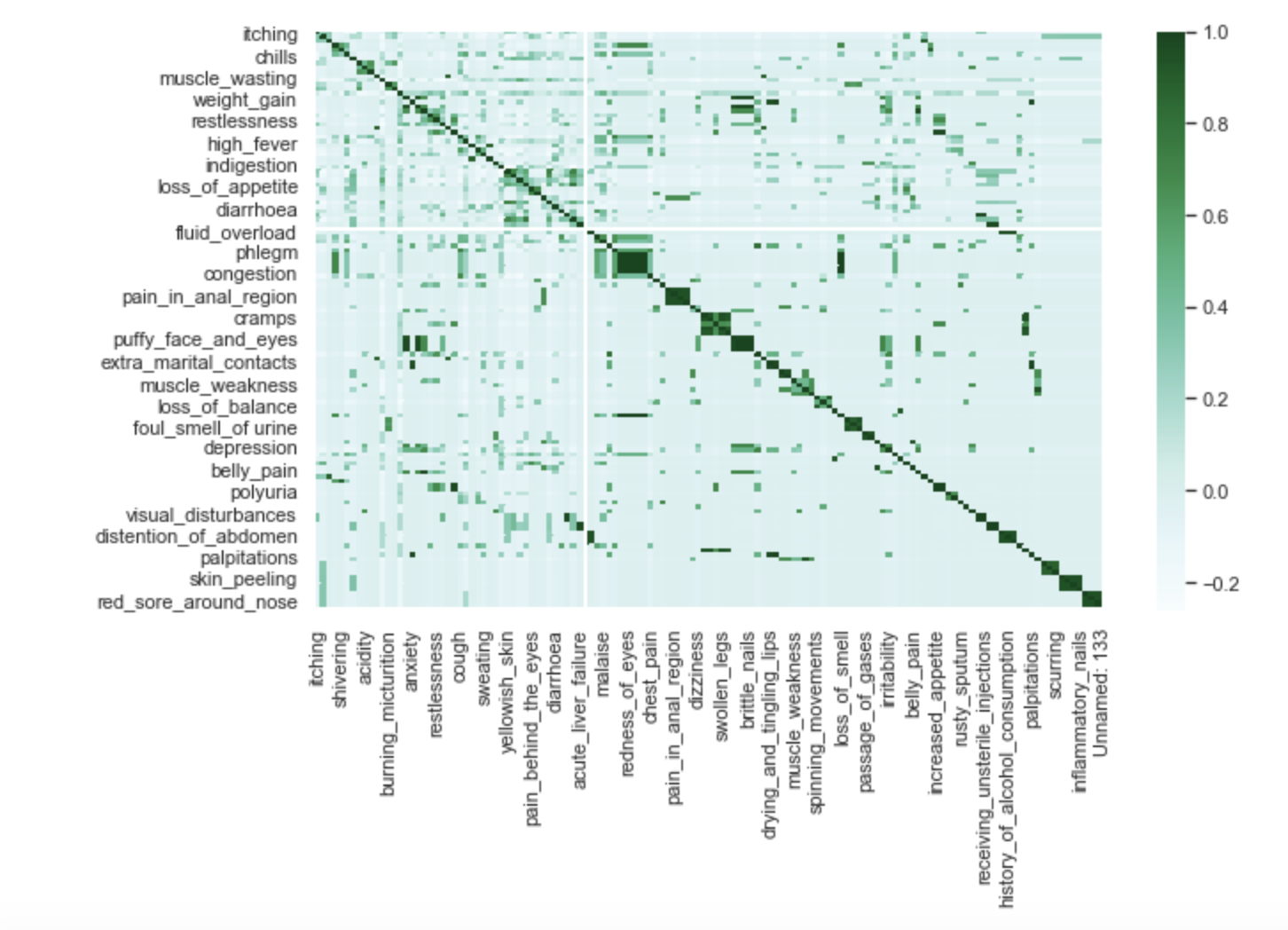


Figure 6: Correlation Matrix of selected features

Similarly all the columns(symptoms) were groupby and looked up prognosis showing those symptoms(having columns value 1) using pie chart to explore the data.

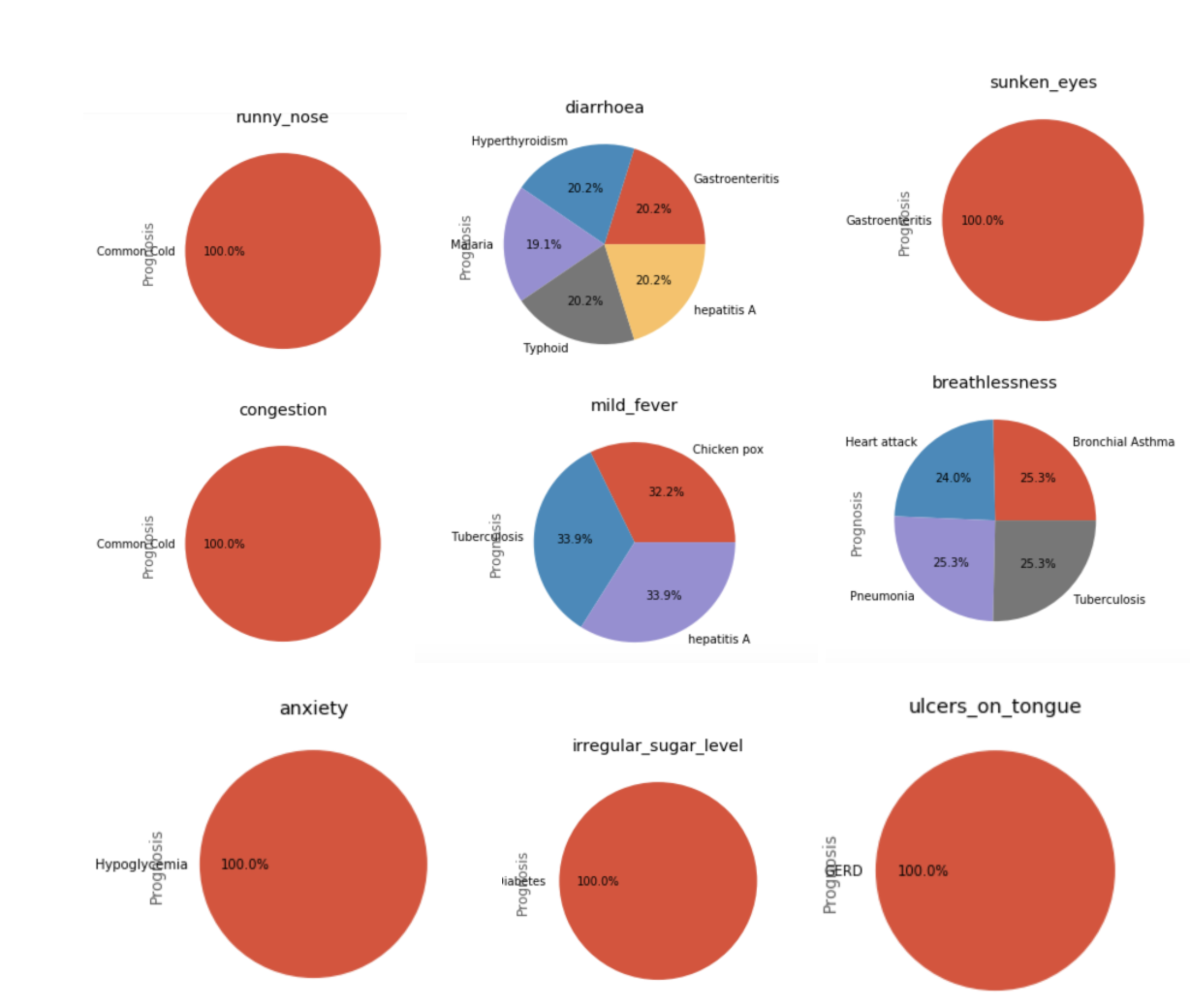


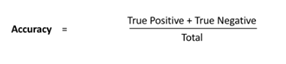
Figure 7: Pie chart of selected features based on prognosis

All the features were kept apart from the last column of the training data set. Data was pretty clean and a little small. All pie charts of specific symptom are attached in the appendix 2.

## Evaluation Metrics

The model will be using various evaluation metrics such as

1. Accuracy: which refers to how close a measurement is to the true value and can be calculated using the following formula



1. Precision: which is how consistent results are when measurements are repeated and can be calculated using the following formula

image

1. Recall: which refers to the percentage of total relevant results correctly classified by the model and can be calculated using the formula

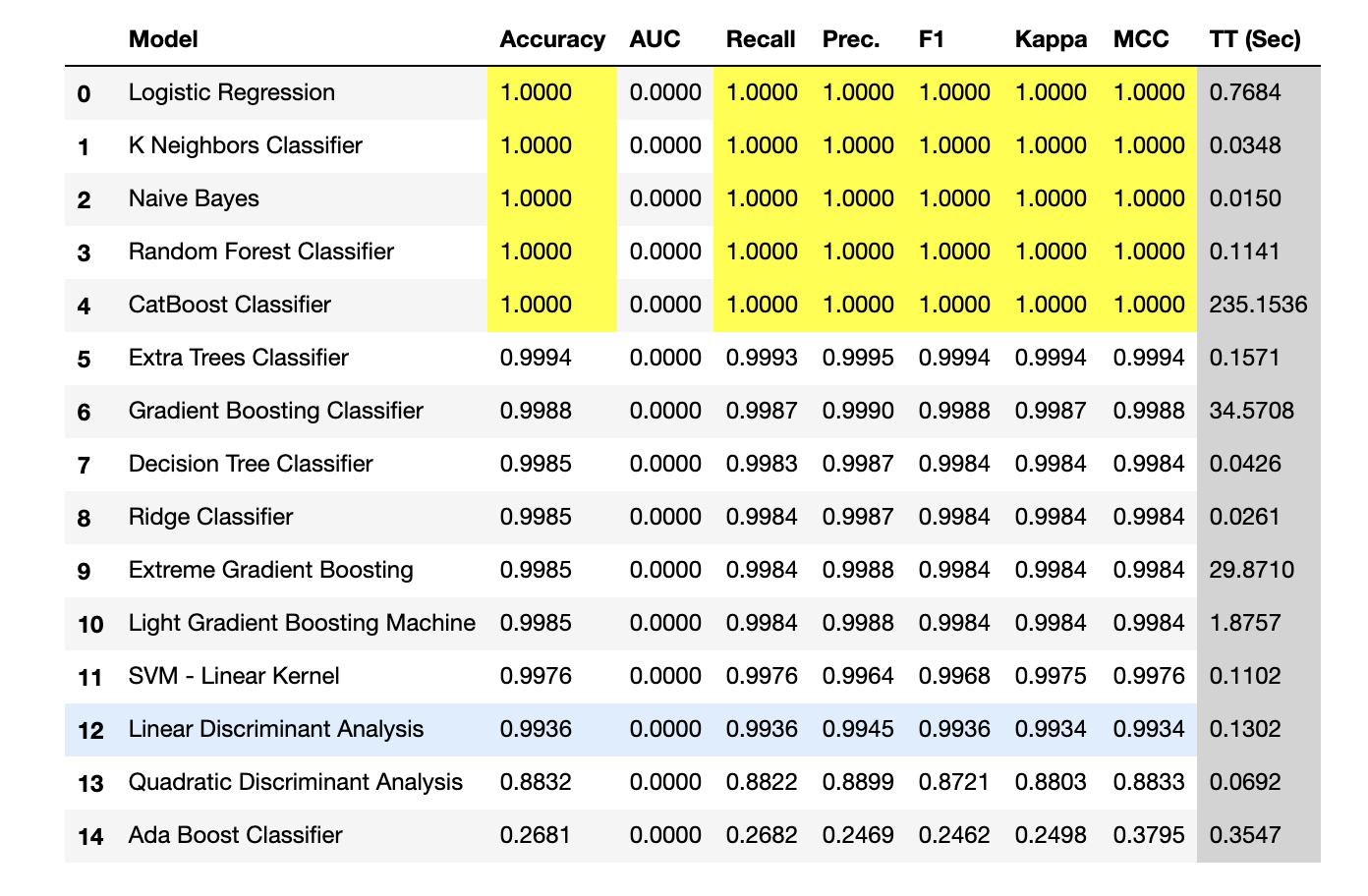
image

# 

# Data Modeling

## PyCaret Classification

PyCaret's classification module was used. This is a supervised machine learning module which is used for classifying the elements into a binary group based on various techniques and algorithms. The below table depicts various classification models and techniques. Using the below as a starting point I want to examine and tune some of the more effective models and find the most important features.

****

Initial impression is that the model is okay. The Logistic regression, K neighbors Classifier, Naive Bayes and Random forest classifier are the best machine learning models for better Accuracy. The Accuracy is 100% which is for all the models mentioned above tested. To evaluate the model we also need to look at the other plots.

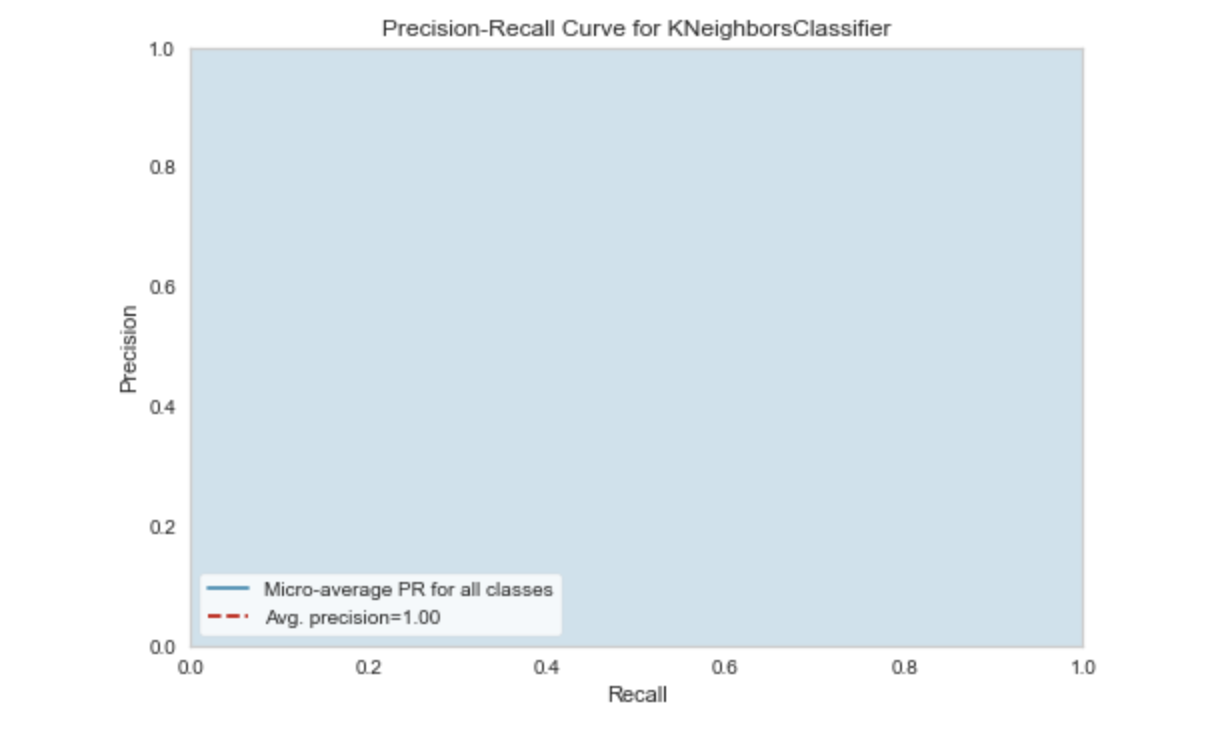


Figure 5: Precision-Recall Curve for KneighborsClassifier

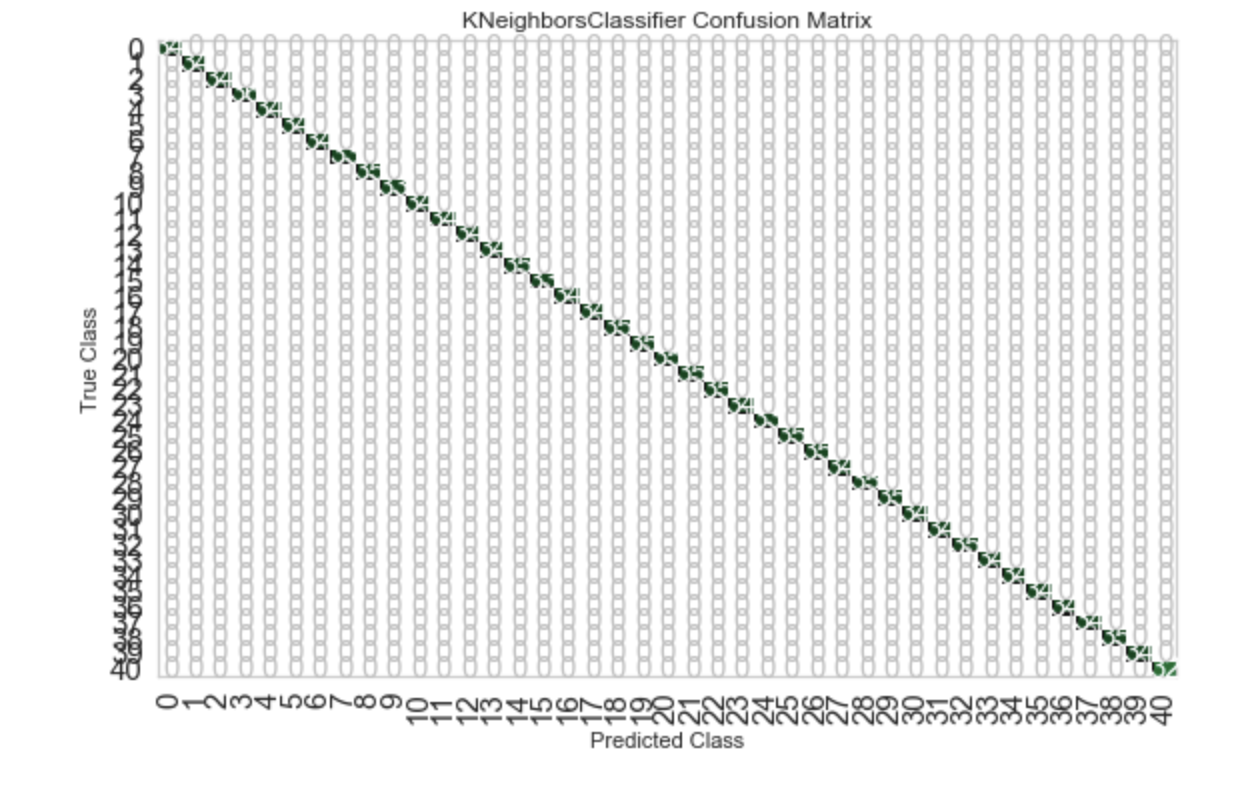


Figure 6: Confusion Matrix for KneighborsClassifier

.

The above confusion matrix shows that.

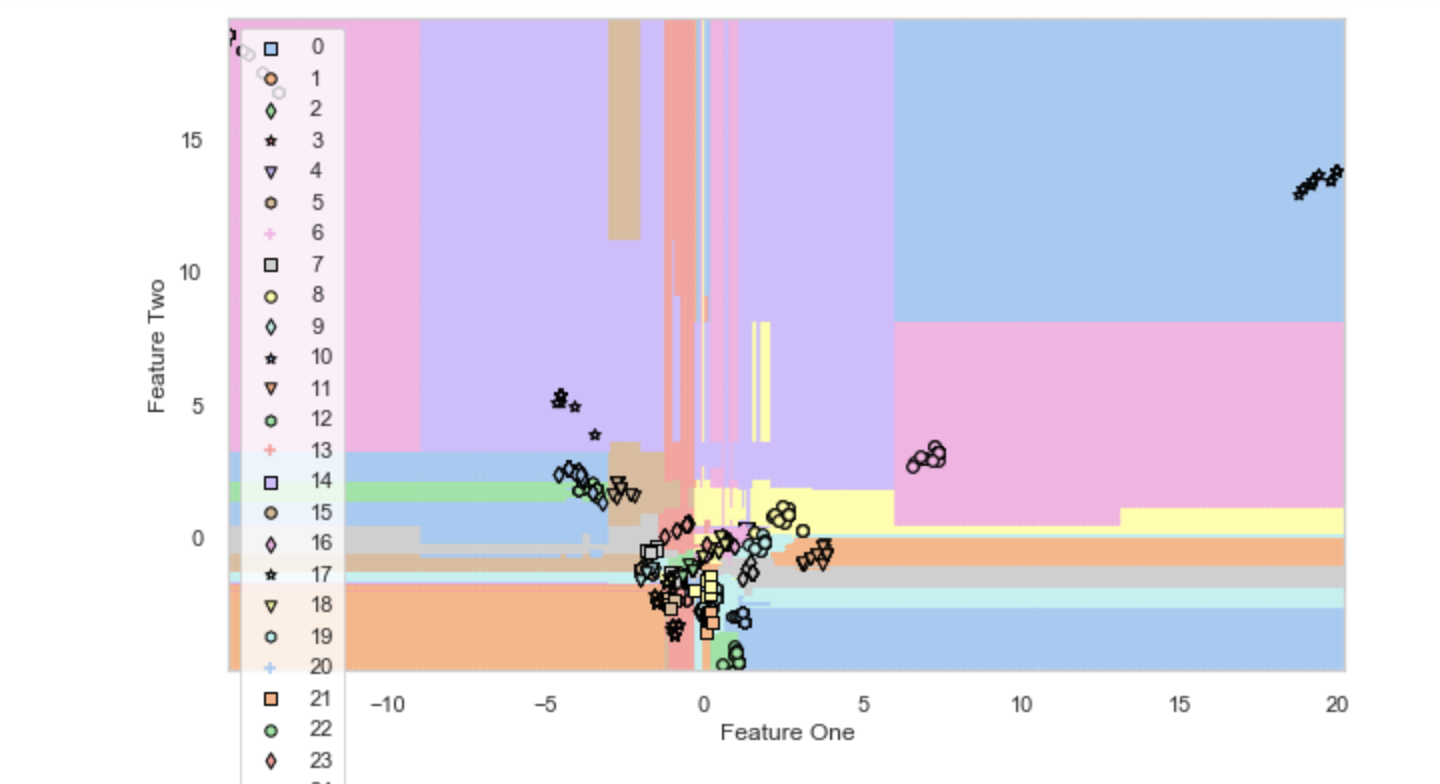


Figure 8: Clusters for KneighborsClassifier

## **k-Fold Cross-Validation**

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

Cross-validation was used for Naive Bayesian model and Kneighbors Classification. Accuracy was 100% for both of them. This identified model is not overfitting.

## **Neural Network**

At the first beginning we tried to add 3-4 hidden layers in our neural network but it performs bad. The test accuracy is only 60%. Then we analyzed that the dataset is not big so we decide to make our network simple. At last we have only 1 hidden layer with 40 neuron nodes. For the optimization we use Adam instead of SGD (Stochastic Gradient Descent) because Adam is a combination of RMSprop and SGD with momentum and it takes advantage of momentum by moving average of the gradient. We got 97.83% accuracy with SGD whereas got 100% accuracy with Adam. Since our dataset is not big, we just choose the batch size to be 200, which is enough for training. And we run 100 epochs to avoid overfitting.

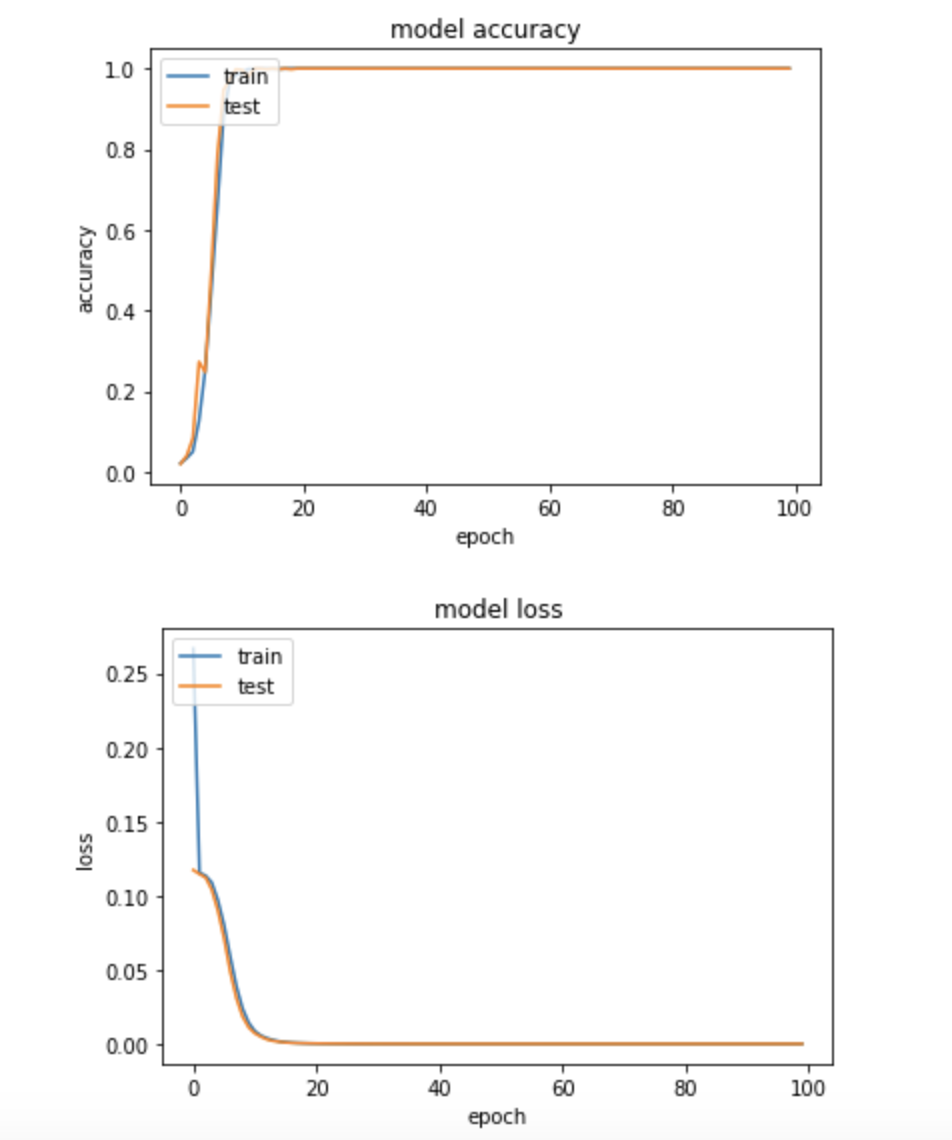


Figure 9:l Accuracy(top) and loss(bottom) of Neural Network model

# **Conclusions**

We use some libraries provided by Python to implement this project. After the experiments, the algorithm of Naïve Bayes,Random Forest, K nearest Neighbors, Logistic regression gives us the best test accuracy, which is 100%. The reason why it outperforms others is that it is not limited to the property of the dataset. Naïve Bayes requires the features to be mutually independent. Logistic Regress requires the features to be linearly separable. SVM requires the parameters to be appropriately set and the neural network requires a complicated and big dataset. We get a good result of 100% accuracy, that is not enough because it cannot guarantee that no wrong diagnosis happens. To better predict, we hope to require more dataset because 4000 instances of dataset are not sufficient to do an excellent job in predicting 41 different diseases.

## **Future Research**

To produce an even more accurate disease prediction model, it would be helpful to obtain a larger dataset as well as a more recent dataset. There are almost certainly medical tests and metrics developed over the last 10 years that would help further improve our identification of disease.

In the future, to predict disease we want to try different diseases such as lung cancer by using image detection. In this way, the dataset becomes complicated and we can apply convolutional neural networks to make accurate predictions.