<u>Disease Prediction Model using Machine Learning Algorithms</u>

CSE445
Section - 3
Spring 2021

Prepared by:

1. **Name: Sumit Kumar Das**
   ID: 1731847642
2. **Tasneem Mahmud**
   ID: 1731893642
3. **Durdana Kamal**
   ID: 1813355042

Abstract

______Every human illness is usually preceded with the display of certain symptoms. The symptoms play a critical role in the eventual onset of a disease. There have been growing efforts to predict disease status, in order to initiate treatment before it gets untreatable. Machine learning has been applied successfully in several biological domains, usually predicting a specific disease. Here, we review the prediction model which will be able to predict a number of diseases, based on their symptoms. We shall review some of the methods and the algorithms that they are based on. We also perform an analysis of symptoms and disease datasets that have given us improved results, using a variety of machine learning and feature extraction methods. The disease prediction model with its code, generated features and datasets are available via GitHub:
https://github.com/durdanakamal/Durdana-Kamal-2-1813355042_CSE445_3_Spring-21

1. Introduction

Supervised machine learning algorithms are heavily used in the data mining field. These algorithms are also useful for prediction models. Medical informatics and disease prediction have been popular topics of discussion among the data science community in recent years, mainly because machine learning has been integrated into the health sector in various forms, such as electronic health records, administrative data, etc.

In this project, we performed One Hot Encoding on the symptoms dataset to "binarize" the category and included it as a feature, to help train the model. We have applied multiple algorithms to train our data, in order to improve our accuracy and predictive prowess. Two databases were taken from www.kaggle.com, which contained disease data and their specific symptoms, and the symptom severity.

Both the Random Forest (RF) algorithm and the Decision tree algorithm showed superior accuracy, precision and F1 scores of 100%.

This project compares between two supervised machine learning algorithms, Decision tree and Random forest, and provides an overview of the predictive performance, precision and accuracy of the model, for disease prediction. This information can provide some insight into the biological field, as it utilises the available information at hand to predict a disease that can be treated as soon as possible.

2. Related Work

The disease Prediction system is based on predictive modeling which will predict the disease of the accused on the assumption of the symptoms that the accuser provides as an entry to the system. The system will then analyze the symptoms given by the user as entry and will give the probability of the disease as an output. The process is carried out using the Decision tree classifier. Decision tree classifier estimates the probability of the disease. With the growth of big data in the health care sector, accurate analysis of medical data leads to early detection of certain diseases, which can then be treated. This system can be used to predict an illness by analyzing the symptoms. A decision tree classifier is used to evaluate the model. This system is used by end-users.This system used the Machine Learning Technology for deciding what diseases the accused patient might have or develop, the decision tree classifier algorithm is used.To improve the accuracy from massive data, the existing work will be done on unstructured data. For the prediction of diseases, the existing will be done on linear, KNN, Decision Tree algorithm.The gain ratio decision tree is used here which is one of the decision tree type .

The gain ratio decision tree is dependent on the information gained approach,which selects the splitting attribute that will not maximize the value of the information gained approach, therefore maximizing the information gain. Information gain is the link between the original information content and the amount of information needed. The features are ranked by the information gained, and then the top-ranked features are chosen as the potential attributes used in the classifier.

To determine the splitting attribute of the decision tree, we must calculate the information gain for each attribute and then select the attribute that will maximize the information gain. [4]

In the case of general diseases prediction using ML where the classification task is used for prediction of working cases dependent on past information. Many data mining techniques such as numpy, neural network, decision tree have been applied by researchers to have a precision diagnosis in heart disease. The accuracy given by different algorithms will vary depending on the number of attributes. This research provides a diagnostic accuracy score for improvement of better health results, as the accused will be informed beforehand.In the particular Heart disease prediction project they have used WEKA tool in this research for pre-processing the dataset. Only 14 out of 76 unique attributes have been considered for analysis purpose to get precise results. Thus by comparison and analysis using different algorithms along with implementation with WEKA tool heart disease can be predicted beforehand and treated early thus increasing the chance of survival of the accused patients.

As there is big data progress in biomedical and healthcare societies, accurate studies of medical data shows that it benefits early disease recognition, patient care and society services. When the measure of medical data is incomplete the precision of study is reduced. Additionally, many regional diseases display symptoms differently across the world, which might reduce the prediction ability of the model. In the particular system, it provides machine learning algorithms for effective prediction of various disease occurrences in disease-frequent societies. [5]

It uses the altered estimate models generated from real-life hospital data. It uses a latent factor model to rebuild the missing data, to deal with the problem of incomplete data. This deals with a regional chronic illness, known as cerebral infarction. It uses Map Reduce algorithm and Machine Learning Decision Tree algorithm on structured and unstructured data from hospitals. As far as we are concerned, none of the existing models or work in the field of medical big data analytics, focused on both structured and unstructured data. As opposed to many other estimate algorithms, the accuracy of the proposed algorithm reaches 94.8% with a faster convergence speed than the CNNbased unimodal disease risk prediction (CNN-UDRP) algorithm.[6]

3. Methodology

In this section, we will first introduce the concept of Support Vector Machine in Section 3.1, where we will elaborate the manner of implementation in our project. Then we will introduce the Random Forest algorithm in Section 3.2, and at last discuss the Decision Tree Algorithm in Section 3.3.

3.1. Support Vector Machine

A support vector machine (SVM) is a machine learning algorithm that can be used for classification and regression problems. It is a supervised learning method that looks at data and categorizes them based on their features.The output of the Support Vector Machine is a map, which contains sorted data. There will be margins between the data and the distance will be maximised. [2]

Applications of SVM include areas such as:

- Image classification
- Text classification
- Hypertext classification
- Protein classification (in biological field)

3.2. Random Forest

Random forest is a machine learning algorithm that gives efficient results even without hyper-parameter tuning. It is a simple, easy-to-use algorithm and it is diverse because it can be used for both regression and classification tasks. [1]

Random forest is a supervised learning algorithm. It builds a “forest” of decision trees, which are usually trained with the “bagging” method. It produces a better result as it is a combination of learning models. The general idea of the bagging method is that a combination of learning models improves the accuracy of the overall result. A random forest builds multiple decision trees and merges them to increase the accuracy in its prediction.

Random forest is diverse as it can be used for both classification and regression tasks. [4]

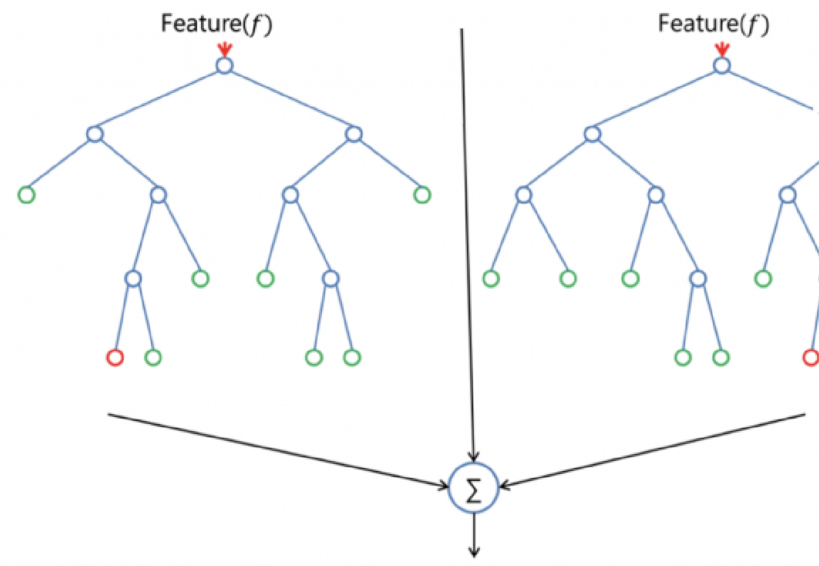


Figure 1. This diagram shows how a random forest would look like with two trees. [1]

Random forest has almost the same hyperparameters as a bagging classifier or a decision tree. Usually there's no need to combine a decision tree with a bagging classifier because you can always use the classifier-class in random forest. By using the algorithm's regressor, we can carry out regression tasks.

Random forest searches for the best feature among a random subset of features, instead of looking for the best feature. This increases the diversity in the results.

To increase the randomness of the model, we can additionally use random thresholds corresponding to each feature rather than search for the best possible thresholds.

While predicting, it is possible to measure the relative importance of each feature. Sklearn provides a tool that measures a feature's importance by looking at how much impurity is reduced across all trees, by the tree nodes that are using that specific feature. The score for each feature is calculated after training and the results are scaled so that the sum of importance equals 1.

Feature importance assists us to avoid overfitting. When there are too many features present, it is easy for the model to pick up on the noise in the data. Feature importance can help eliminate some features that have little to no importance.

The hyperparameters of sklearns built-in random forest function, can increase the predictive power and speed of the model.

The n_estimators hyperparameter is the number of trees that the algorithm will build before taking the average prediction. The higher the number of trees, the better the accuracy. However, the computation time will also increase.

Secondly, another hyperparameter is max_features, which determines the maximum number of features that random forest will consider in order to split a node.

Thirdly, another hyperparameter is min_sample_leaf, which is the minimum number of leafs that are required to split an internal node.

The n_jobs hyperparameter gives the engine the information of the processor usage limit. A value of 1 means it is allowed to use only one processor. A value of -1 means it is allowed to use infinite processors, i.e. there is no limit.

The random state hyperparameter makes sure that the result of the model is repeatable. The model will always provide the same predictive results if a random state is fixated on a specific value and if the model has to work on the same training data and hyperparameters.

Lastly, there is a cross-validation method oob score (which might also be called oob sampling). About one-third of the data is not used as training data in this sample, but can definitely be used to assess the model’s performance. They are called out-of-bag samples. It can be compared with the leave-one-out cross-validation method, except the advantage is that there is no extra computational work involved.

3.3 Decision Tree

One of the earliest and most popular machine learning techniques is the decision tree (DT). A decision tree represents the decision logic in order to classify data objects into a tree-like structure, i.e. tests and outcomes. A decision tree usually has many layers of nodes, and the root node is usually at the top of the tree. [1]

All the internal nodes (with at least one child) represent input variable or attribute testing. The algorithm branches towards the appropriate child node depending on the test result, and the procedure of testing and branching takes place over and over again until it reaches the leaf node.

The results of each choice are shown by the terminal or leaf nodes. DTs are frequently implemented in many medical diagnostic systems, as they are very straight-forward and do not require much time to comprehend and learn.

During sample classification, we traverse the decision tree. The outcomes of all the tests at each node along the traversed path, will supply enough information to make a prediction about the class of the sample. [3]

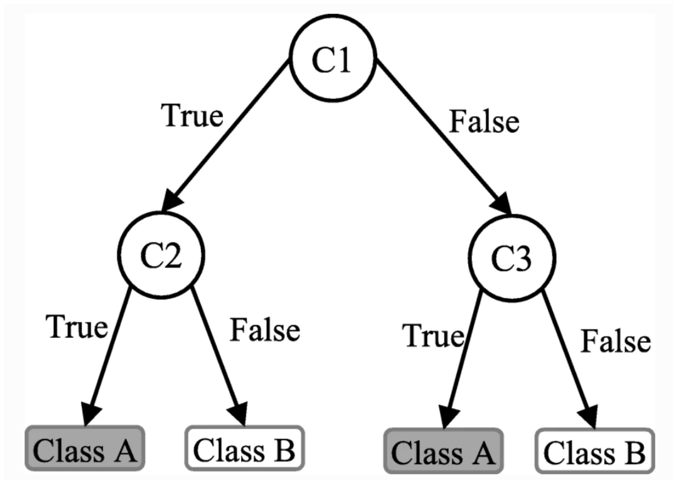


Figure 2. **A decision tree. A circle represents each variable (C1, C2, and C3), whereas rectangles represent the different outcomes based on your choice (Class A and Class B). Each branch says 'True' or 'False' based on the**

outcome value from the test of its individual parent node. This is how we can classify a sample to a class. [1]

The decision tree model makes analysis based on three nodes.

A. The main node is called the root node, which is what determines the other nodes' functions.
B. The interior nodes deal with numerous attributes.
C. The leaf nodes show the test results

Depending on the most crucial indicators, the DT will split the data into two or multiple numbers of sets. Predictors can either have the most gain of information or a minimum value of entropy, which we will initially have to calculate using the formula below:

Entropy(S)=∑i=1c−Pilog2Pi,

Gain (S,A)=Entropy(S)−∑v∈Values(A)|Sv||S|Entropy (Sv). [4]

3.4. Difference between random forest and decision trees

When you give a decision tree a training dataset with features and labels, it will generate a set of rules that will be used to predict.

The random forest algorithm, on the other hand, chooses analyses and features randomly to produce several decision trees and then calculates an average of the results.

Another difference is that "deep" decision trees may be susceptible to overfitting. Random forest usually avoids this by producing random subsets of the different features and using those selections to build smaller trees. There is a slight disadvantage, however, as the greater the number of trees, the larger the computational time and hence the smaller the chances of the predictive model to work.

4. Results

We used a database from Kaggle for this project. Here is the link https://www.kaggle.com/itachi9604/disease-symptom-description-dataset?select=symptom_Description.csv&fbclid=IwAR2DAnvYMce_GAkzxtJtb8HOb-4EuexTvluPh_N3ovk-_tnINajaMVNIDaA. Here, we used two datasets one called “dataset.csv” and the other is “Symtom-severity.csv”. We mainly used the libraries numpy and pandas. Firstly, we used .head() to see what the dataset looks like. It showed the first five rows as default. After taking a look at the dataset we saw that there were a lot of null values starting from Symptom7 to Symptom17. Since, there were so many null values we decided to get rid of those columns and ended up with six symptom columns. Then we used .describe() to check different attributes of the dataset. Afterwards we check for null values in the remaining columns. We saw that symptom4, symptom5 and symptom6 still had some null values. So we filled the null blocks with 0 later on in the code. Next we found the count of unique items in each column.

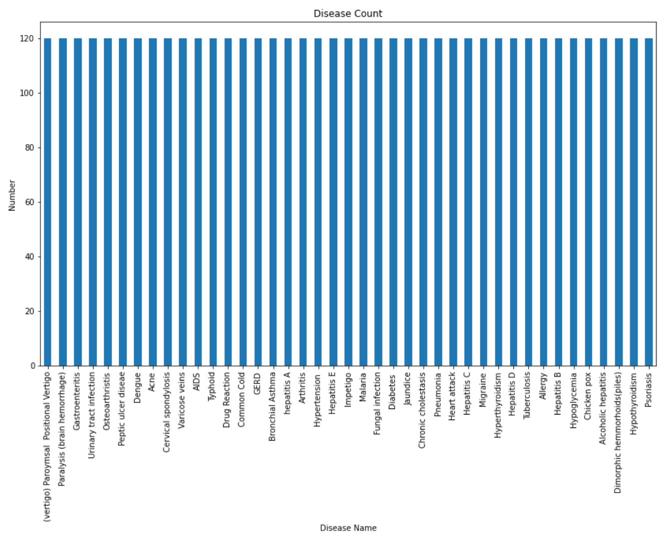


Figure 3. Disease Count

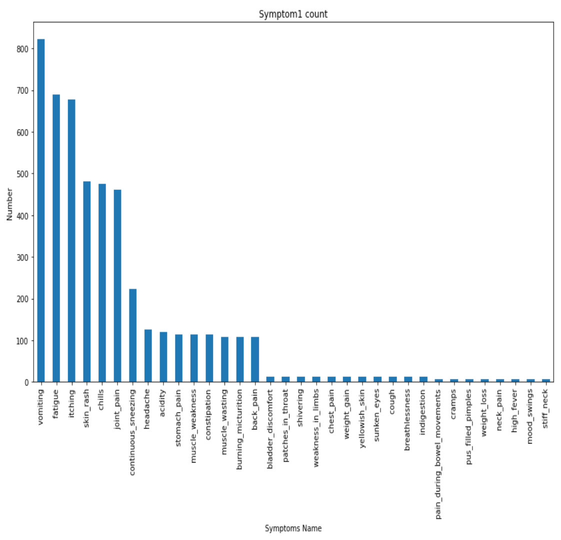


Figure 4. Symptom 1 Count

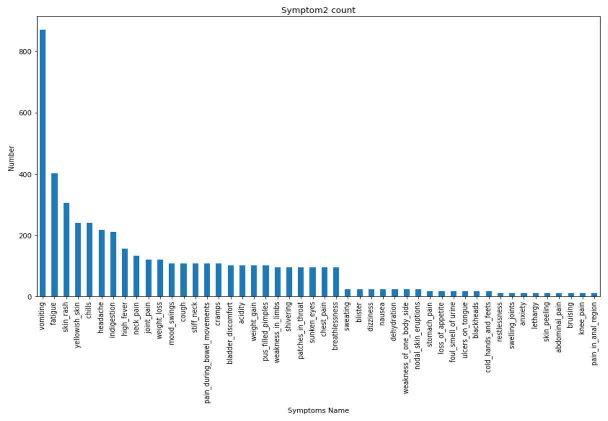


Figure 5. Symptom 2 Count

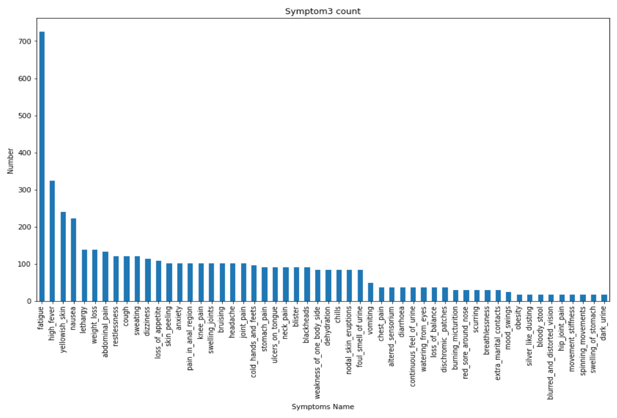


Figure 6. Symptom 3 Count

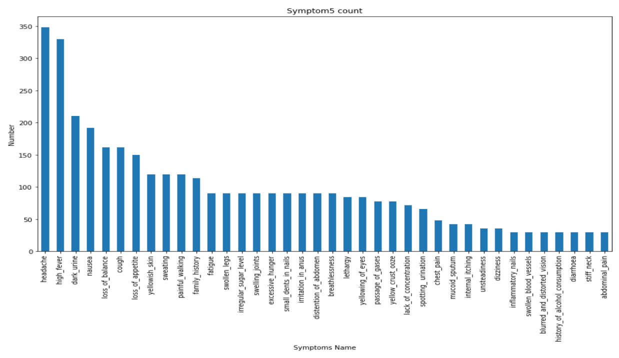


Figure 7. Symptom 5 Count

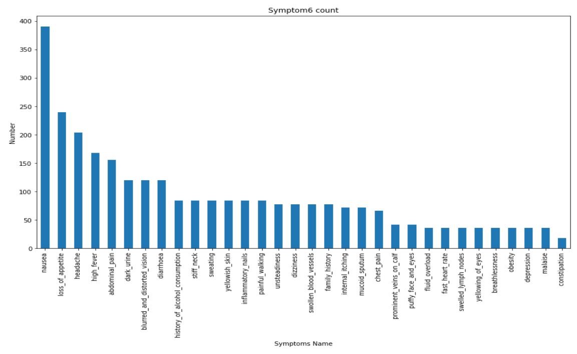


Figure 8. Symptom 6 Count

Then we removed extra spaces from the data in the columns to make the data consistent. Then we created relplot diagrams to check the matching symptoms against the diseases.

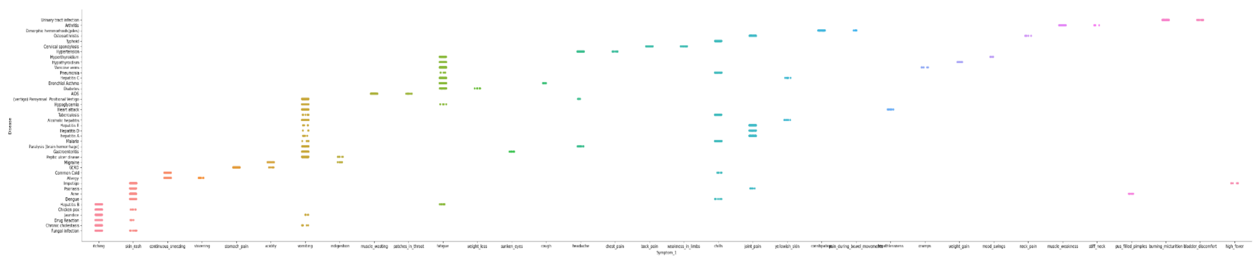


Figure 9. Disease against Symptom 1

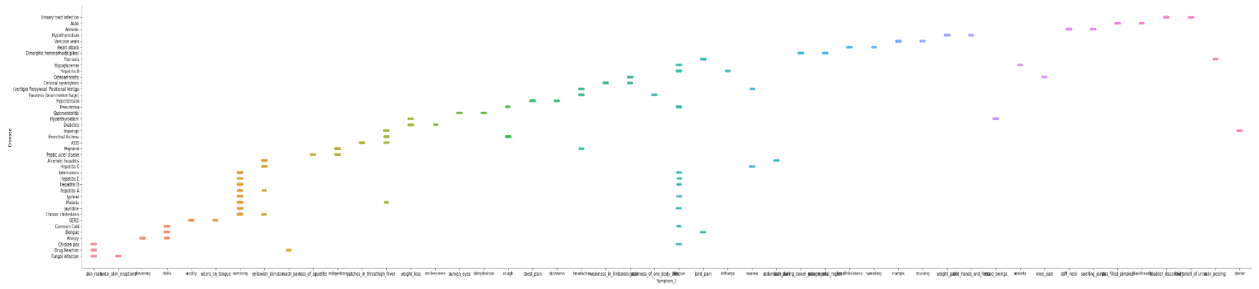


Figure 10. Disease against Symptom 2

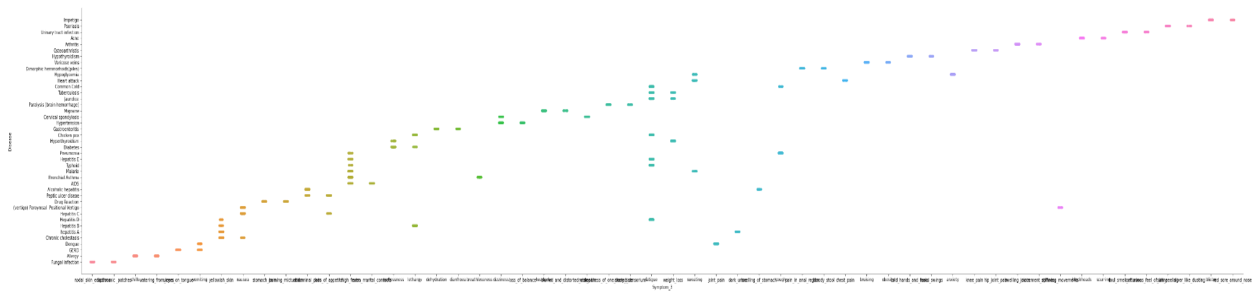


Figure 11. Disease against Symptom 3

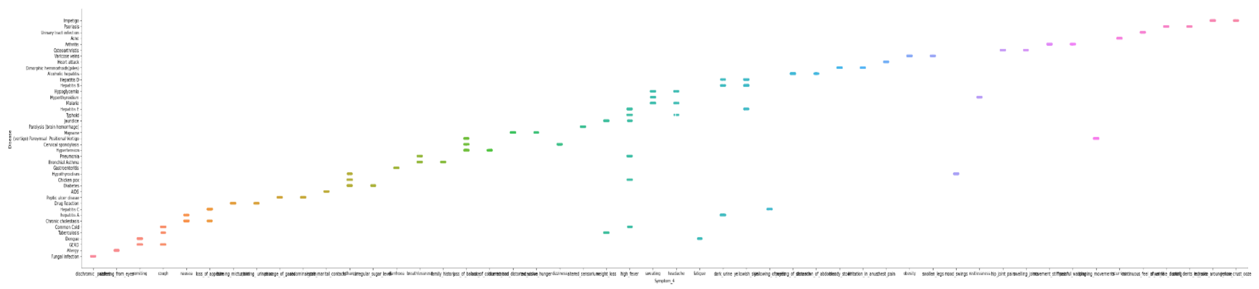


Figure 12. Disease against Symptom 4

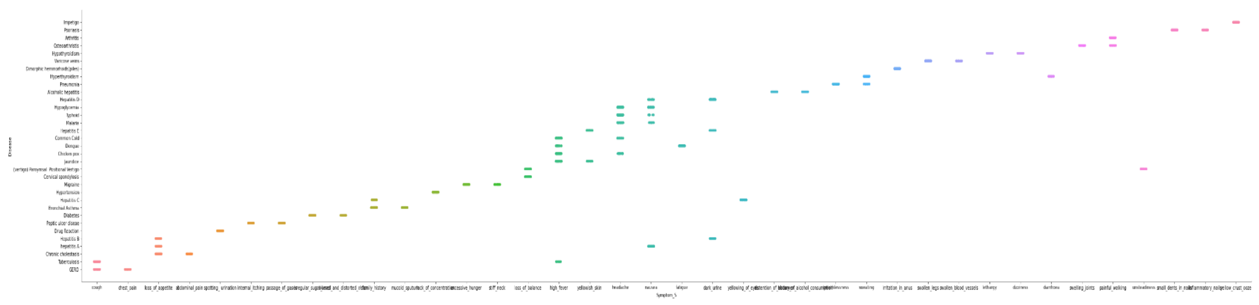


Figure 13. Disease against Symptom 5

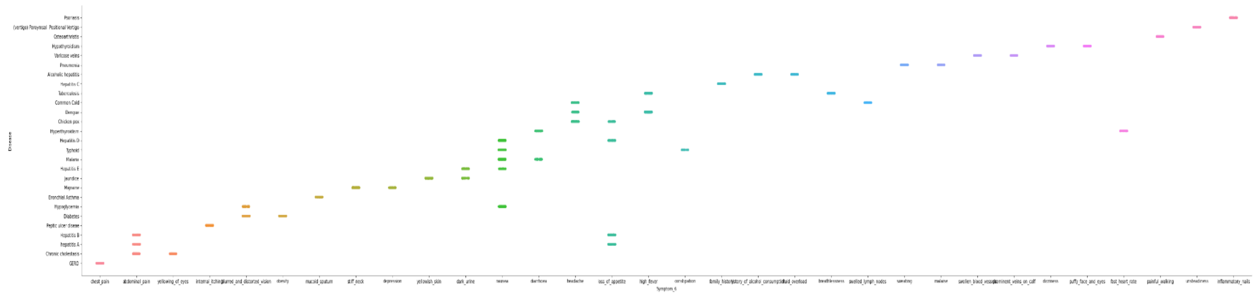
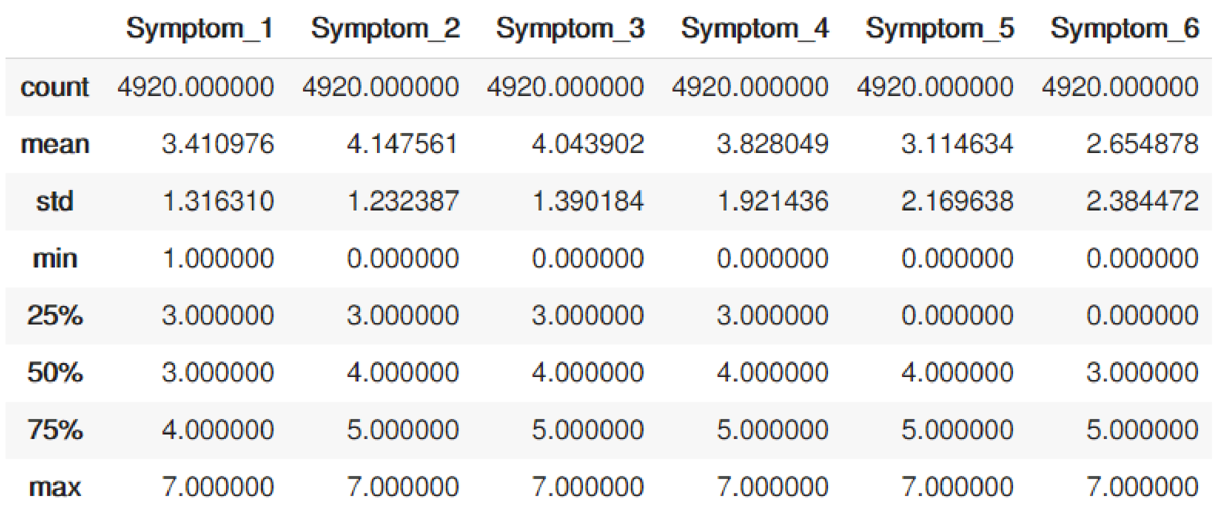


Figure 14. Disease against Symptom 6

Then we integrated the two datasets and replaced the symptoms by their severity. After replacing the symptoms we saw that there were some symptoms that did not have any weight. So we replaced those symptoms with 0. And after replacing the symptoms with numerical data we used .describe() to check the mean, mode, etc.



	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6
count	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000	4920.000000
mean	3.410976	4.147561	4.043902	3.828049	3.114634	2.654878
std	1.316310	1.232387	1.390184	1.921436	2.169638	2.384472
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	3.000000	3.000000	3.000000	3.000000	0.000000	0.000000
50%	3.000000	4.000000	4.000000	4.000000	4.000000	3.000000
75%	4.000000	5.000000	5.000000	5.000000	5.000000	5.000000
max	7.000000	7.000000	7.000000	7.000000	7.000000	7.000000

Table 1. Data description

Then we checked for matching symptoms in each column using .pairplot().

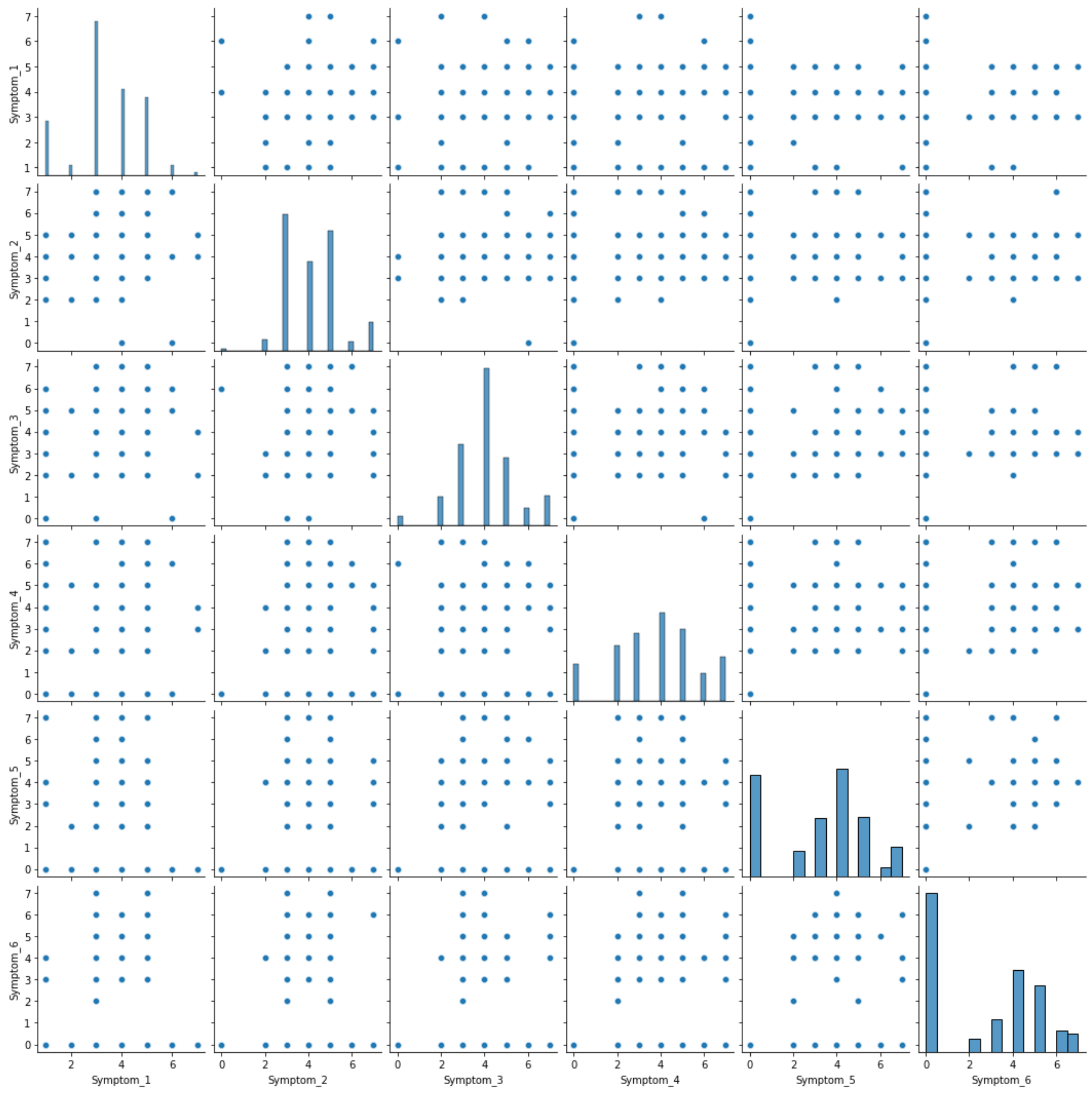


Figure 15. Checking matching symptoms in each Symptom column

We then created a correlation matrix using sns.heatmap()

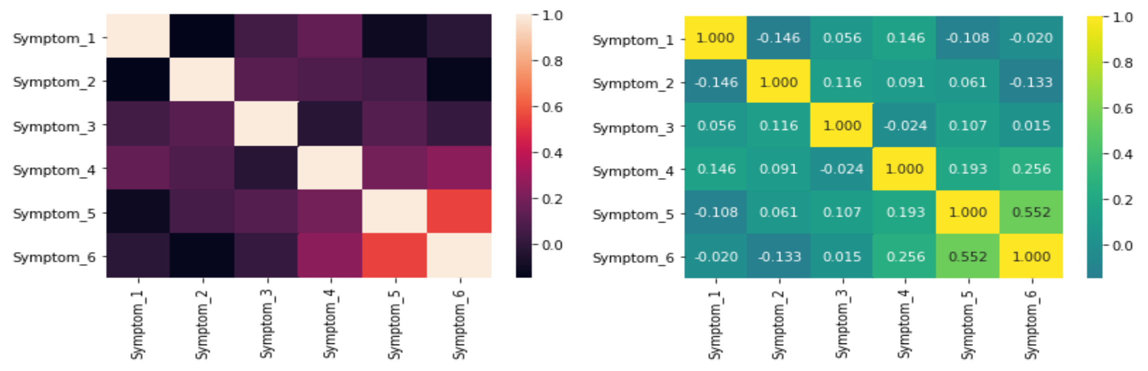


Figure 16. Correlation matrix

Then we split the data taking the disease column in a different variable. After that we performed one hot encoding since the disease column was completely categorical. Then we used the SVM model to train and check accuracy. We got a f1 score of 96.99%, Accuracy of 96.99%, Precision 97.07% and Recall 97.17%. Then we created a confusion matrix for that model.

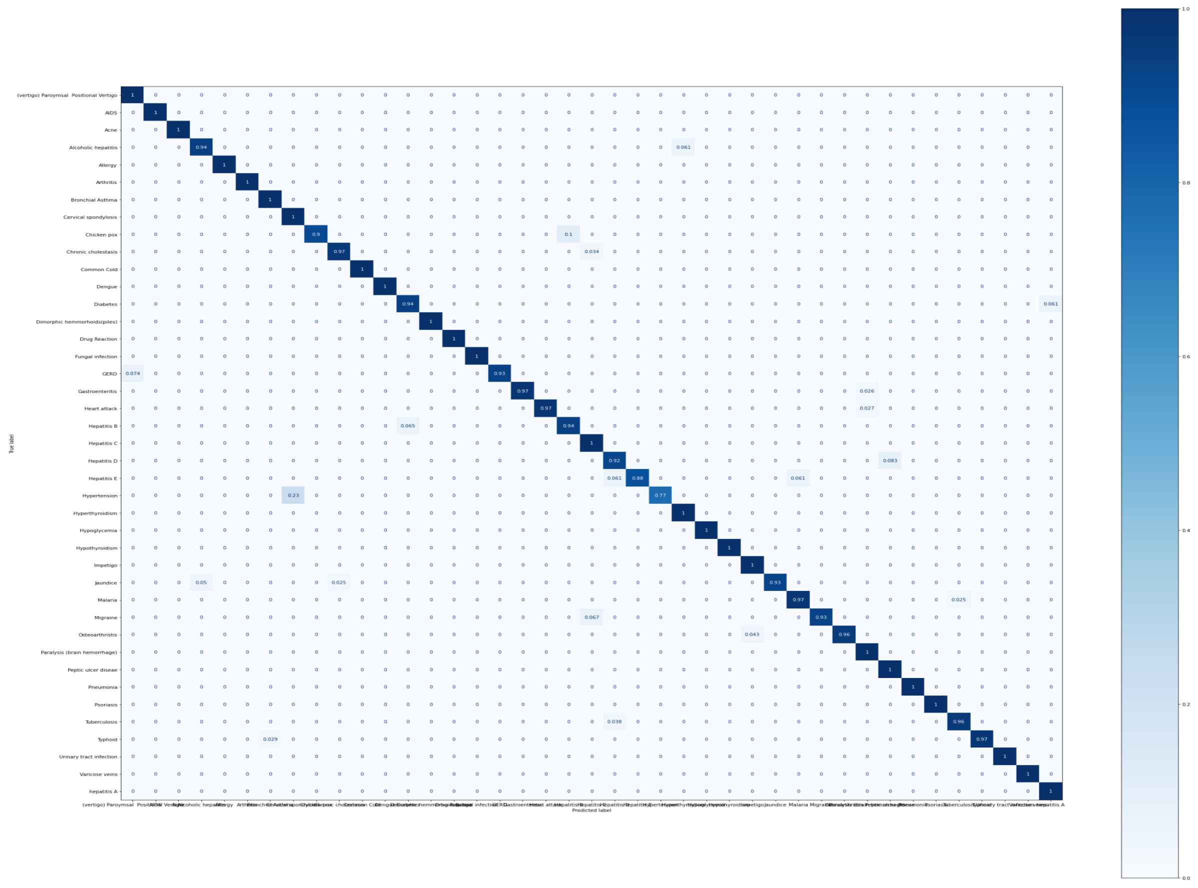


Figure 17. Confusion matrix for SVM

Then we used the Random Forest Classifier. For this model F1,Accuracy,Recall and Precision all were 100%. Here is the confusion matrix.

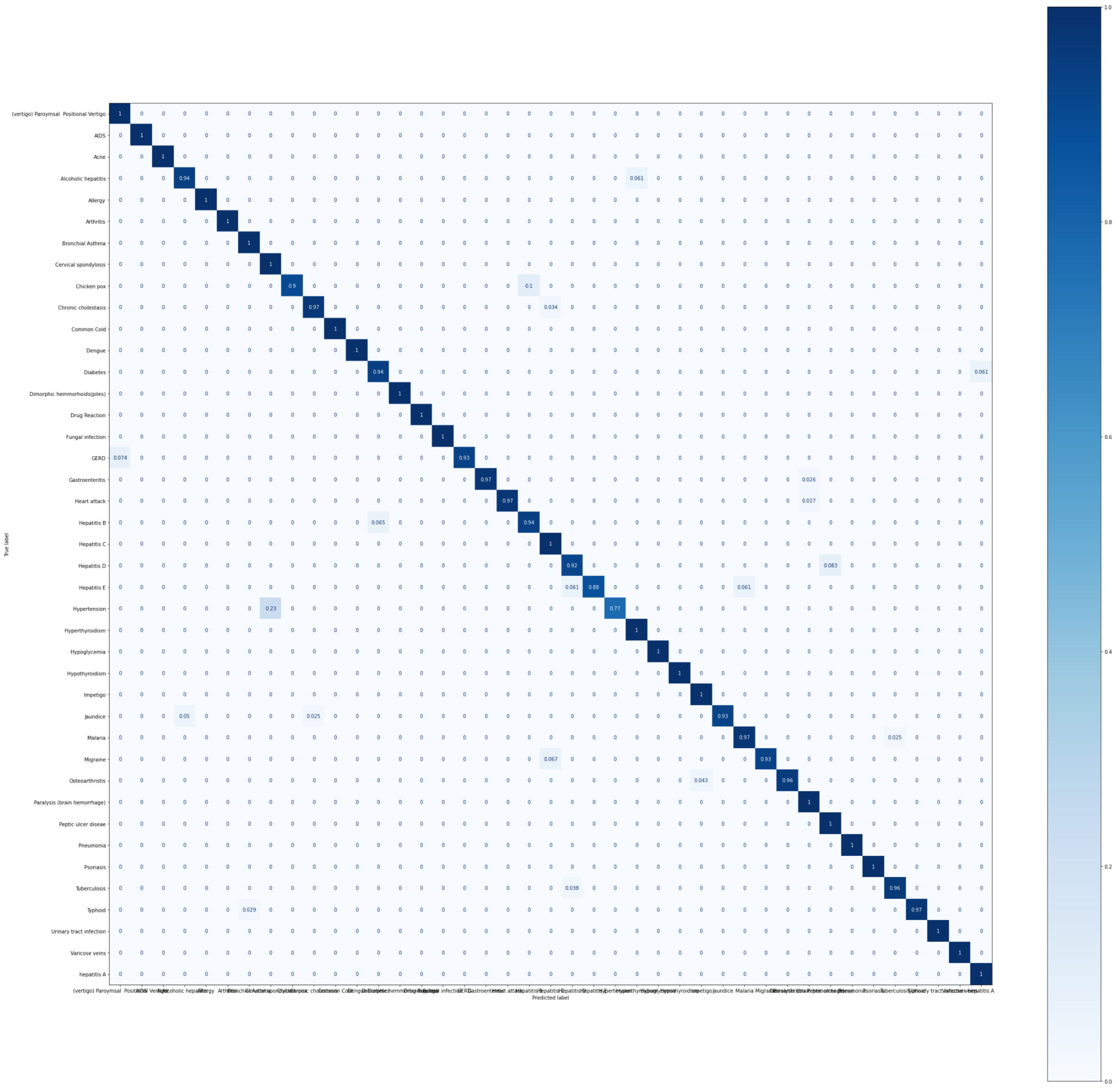


Figure 18. Confusion matrix for Random Forest Classifier

Lastly, we used a Decision Tree. For this F1 Accuracy, Precision and Recall all were 100%. Here is the confusion matrix.

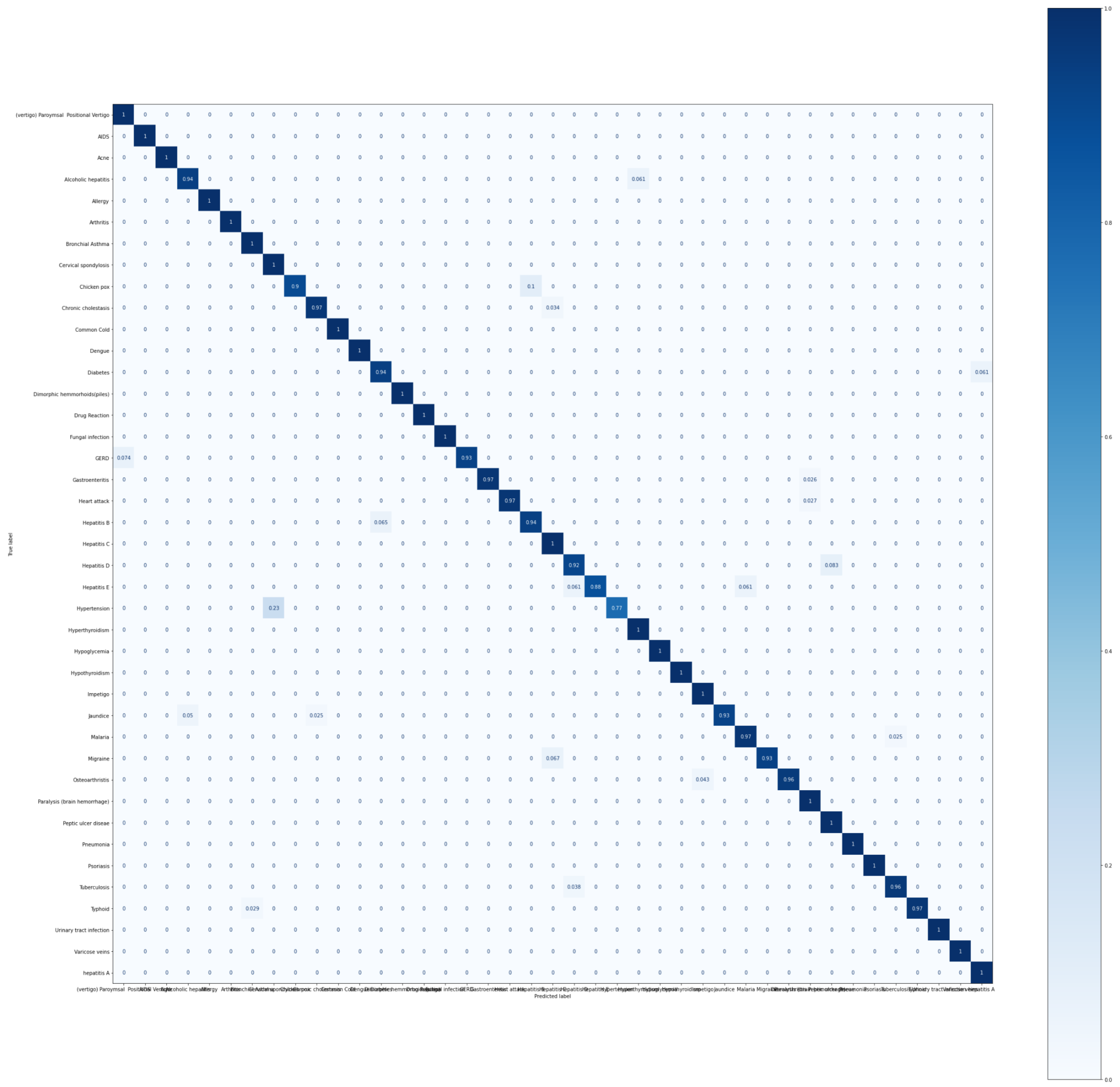


Figure 19. Confusion matrix for Decision Tree

5. <u>Conclusion:</u>

Illness isn’t something people take lightly. Often the panic is more harmful than the disease itself. Especially when an unknown life threatening disease is spreading worldwide so quickly that it gives us flashbacks of the 1910s, when the destructive wave of the spanish flu created havoc all over the world. This virus, widely known as the coronavirus, has already taken millions of lives and has left the rest in panic. Our aim is to help people understand the disease & its significance clearly, so that they can save their own lives and of those they care about, by taking necessary precautions. And in the future, no one should live their lives oblivious to the diseases out there. A timely prediction of a disease will help to slow down the disease, if not prevent it. The main aim of this research is to determine whether or not a patient will develop any kind of disease or not. This research was done using Numpy, decision tree, random forest which are on supervised machine learning classification techniques . Research was performed on 6th generation Intel Corei5 having an 6200H processor up to 2.4 GHz CPU and 8 GB ram. After classification of the dataset, we split it into a training set and a test set. Pre-processing of the data is done and Numpy, decision tree, K-nearest , and random forest are applied to get an accuracy score which are supervised classification techniques such as . For the training and validation data sets, we used Python programming to calculate the accuracy results for the several classification techniques. Percentage accuracy scores are depicted in different algorithms.

6. Acknowledgement

We are thankful to our respectable teacher, Mr. Intisar Tahmid Naheen , whose guidance was crucial for us to execute this project. Thank you, sir, for always being available to discuss problems at great lengths.

References:

[1] Uddin, S., Khan, A., Hossain, M. *et al.* Comparing different supervised machine learning algorithms for disease prediction. *BMC Med Inform Decis Mak* 19, 281 (2019). https://doi.org/10.1186/s12911-019-1004-8

[2] https://www.techopedia.com/definition/30364/support-vector-machine-svm

[3] Raj H. Chauhan, Daksh N. Naik, Rinal A. Halpati, Sagarkumar J. Patel, Mr. A.D.Prajapati-"Disease Prediction using Machine Learning",May 2020,

https://www.irjet.net/archives/V7/i5/IRJET-V7I5385.pdf

[4] Devansh Shah, Samir Patel & Santosh Kumar Bharti -"Heart disease prediction." *SpringLink*, October 2020,https://link.springer.com/article/10.1007/s42979-020-00365-y

[5] S, Vinitha and S, Sweetlin and H, Vinusha and S, Sajini, Disease Prediction Using Machine Learning Over Big Data (February 2018). Computer Science & Engineering: An International Journal (CSEIJ), Vol.8, No.1, February 2018- https://ssrn.com/abstract=3458775 or http://dx.doi.org/10.2139/ssrn.3458775

[6] Nishmitha K Uchil, Dr. M Sharmila Kumari, 2019, Cerebral Infarction Prediction by Machine Learning Over Big Data, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) RTESIT – 2019 (VOLUME 7 – ISSUE 08),