AI Project Report – Virtual Doctor

Taha Khan : K17-3753

Ali Ammar Khan : K17-3713

Safi Ullah Kharadi : K17-3737

Section : B

Submission Date : 10h May, 2020

Submitted To : Dr. Fahad Sherwani

**Abstract:**

This web application was created to improve and provide a digitized way of communication between and doctor and patient, and use machine learning on the data provided for more effective diagnosis of the underlying conditions. This application allows the patients to book appointments, view prescription and reports, manage bills etc. On the other hand a doctor use this application to manage his/her appointments, check patient’s previous medical history and report, prescribe prescriptions etc. In addition to effective management and storage of data, this application also uses supervised learning algorithms to predict diseases on the basis of the symptoms provided by the patient. These symptoms are provided via a chat bot which communicates with the patient and get his/her symptoms. This applications also predicts the possibility of heart diseases of the patients on the basis of their heart report. The algorithms used to solve these classification problems are decision trees and KNN respectively. Use of machine learning provides the doctors a better framework for treating patients.

**Introduction:**This application is a platform for making doctor-patient interaction easier. The system will be used to get the information from the patient and then storing that data for future usage. Doctors normally rely on the information provided by the patient and the previous reports provided by them. Sometimes due to some lack of communication or some miscommunication, and by the lack of information provided by the patient, the doctor is to unable to provide the necessary health care to the patient. The intention of our system ease the interaction between the two with proper storing of medical history, tracking patients previous prescription, provide the doctor with ALL the previous medical interactions and a proper appointment system for the patient. Now to facilitate the doctors in diagnosing patients, this application uses machine learning to provide the doctors with possible diagnosis, so that they have some initial idea of the problem and the patient could get a better and faster treatment. This application uses two machine learning models to perform this task. First on the basis of the symptoms provided by the patient through communication with the chat bot, we use a decision tree classifier to predict the possible disease. This input in this case will the symptoms of the patient {weight loss, skin rash, cough etc.} and output will be the disease for e.g. ‘Jaundice’ .Second according the heart report of the patient, we use a K-nearest neighbor model to predict whether the patient has a possibility of a heart disease or not. The input in this case is the heart report of the patient {age , testbps, chol etc.} and output is ‘yes’ for a possible heart disease and ‘No’ for not.

**Related Work:**

[1].In this blog, the author uses the same dataset as used in this project for heart disease prediction. The author uses the following classification models

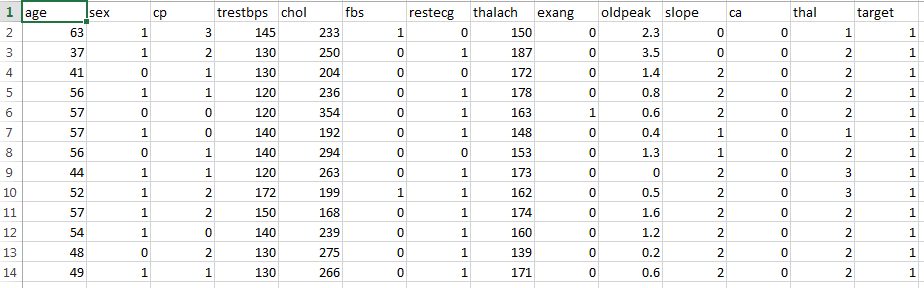
* SVM
* Naïve Bayes
* Logistic Regression
* Decision Tree
* Random Forest
* LightGBM
* XGboost

The highest accuracy is achieved by Logistic Regression and SVM which is 80.32% on the test set. The problem with this implementation is there is not much data preprocessing performed, and also the models are just tested on training and test sets, which are too dependent on the type of split done. A better approach would have been to use cross validation, it would have given a better picture of the model’s performance.

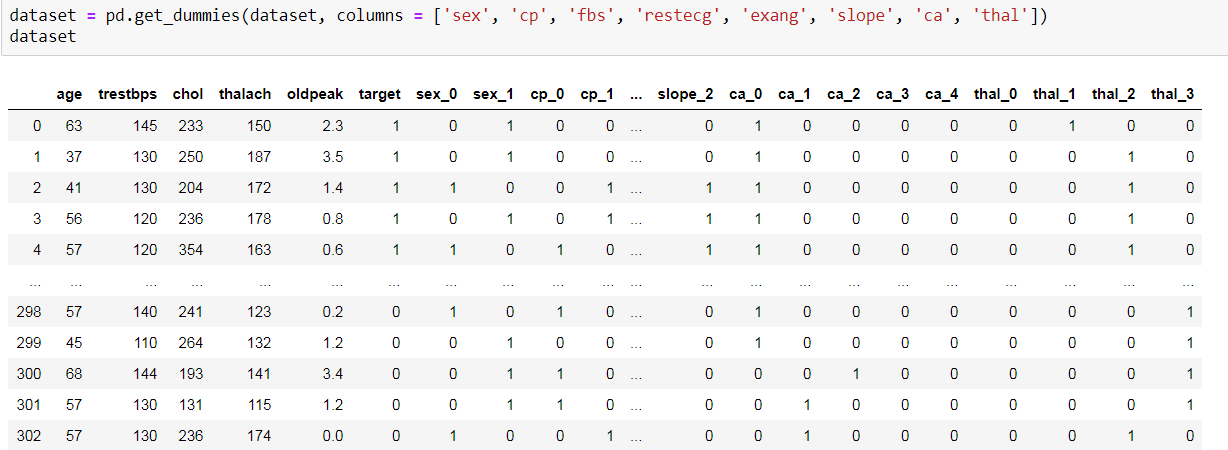
[2]. for disease prediction, this research paper solves a similar problem to ours using K-mean algorithm. This implementation incorporate the symptoms collected from multisensory devices and other medical data and store them into a healthcare dataset. This dataset would then be analyzed using K-mean machine learning algorithms to deliver results with maximum accuracy.

**Dataset and features:**

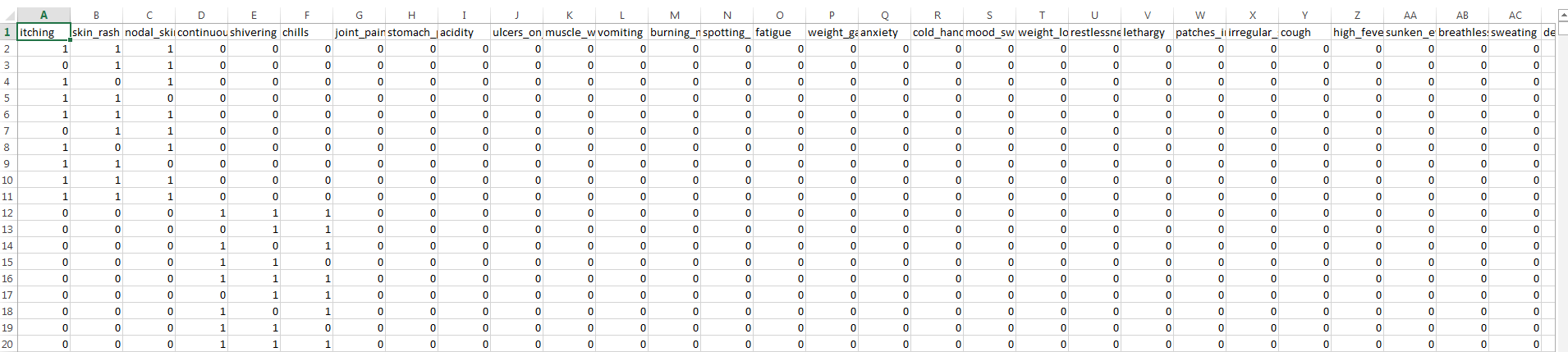
The heart disease data was obtained from kaggle (<https://www.kaggle.com/ronitf/heart-disease-uci>) and consists of 13 features and a target. The target has binary values 0 or 1.

****

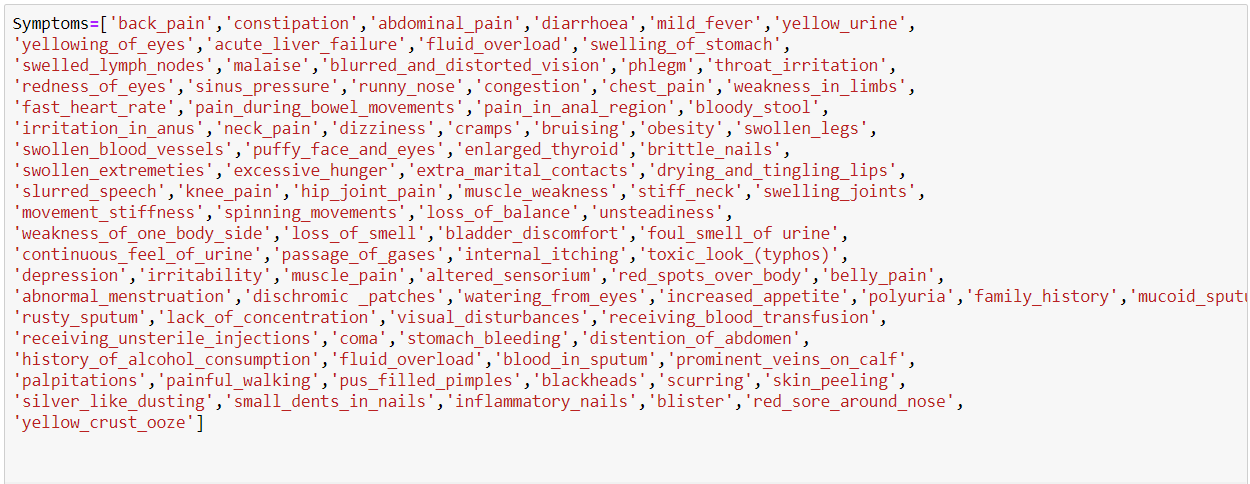
After exploring the data set, for a better model performance, first we converted the categorical variables using the get dummies method to create dummy columns for categorical variables. Then if we wish to use algorithms like KNN it is necessary to scale our features so they may contribute equally since these models use the distance between observations calculated from the features. So we used StandardScaler method from sklearn.preprocessing to enforce that the standard deviation of the features is 1.



For Disease prediction using symptoms the dataset was obtained from kaggle (<https://www.kaggle.com/neelima98/disease-prediction-using-machine-learning>). The data originally had 132 features and 1 target variable.



Since the dataset has a lot of features, some symptoms may not have much effect on the prediction. The same accuracy can also be obtained using a reduced number of features. So for selecting a subset of features from the original features, we used greedy forward stepwise algorithm. The following subset of features (95) were extracted from the original:



**Methods:**

We tested several machine learning models on these classification problems and measured them according to their accuracies (Details in the next section). The models that were selected are

* K-nearest neighbor for heart disease prediction
* Decision trees for disease prediction according to symptoms.

These models were implemented using sklearn library in python. The API for our machine learning models was created using Flask, while rest of our backend work was done using nodejs, and front end was made using React. Following are details of the algorithms, methods used in this project.

**K nearest neighbor:**

Let m be the number of training data samples. Let p be an unknown point.

1. Store the training samples in an array of data points arr []. This means each element of this array represents a tuple (x, y).
2. for I = 0 to m:
3. Calculate Euclidean distance d (arr[i], p).
4. Make set S of K smallest distances obtained. Each of these distances corresponds to an already classified data point.
5. Return the majority label among S.

**Decision Tree:**

Assumptions we make while using Decision tree:

* At the beginning, we consider the whole training set as the root.
* Attributes are assumed to be categorical for information gain and for gini index, attributes are assumed to be continuous.
* On the basis of attribute values records are distributed recursively.
* We use statistical methods for ordering attributes as root or internal node.
* Pseudocode :
* Find the best attribute and place it on the root node of the tree.
* Now, split the training set of the dataset into subsets. While making the subset make sure that each subset of training dataset should have the same value for an attribute.
* Find leaf nodes in all branches by repeating 1 and 2 on each subset.

**Pseudocode** :

* Find the best attribute and place it on the root node of the tree.
* Now, split the training set of the dataset into subsets. While making the subset make sure that each subset of training dataset should have the same value for an attribute.
* Find leaf nodes in all branches by repeating 1 and 2 on each subset.

While implementing the decision tree we will go through the following two phases:

Building Phase

* Preprocess the dataset.
* Split the dataset from train and test using Python sklearn package.
* Train the classifier.

Operational Phase

* Make predictions.
* Calculate the accuracy.

**Feature Selection:**

We used a greedy forward stepwise algorithm for feature selection in disease prediction. The algorithm is given below:

Let S be set of selected features and F be the set of all the features

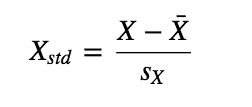
S = null

For all features in F

1. Select a feature of from the set F
2. For features S U {f} , test the accuracy of model on test set
3. If the accuracy of the model is better than previous then add the feature in S
4. Else drop the feature
5. Repeat step 1-4 for all features in F

**Standardization:**

For standardization of features in heart disease dataset, the following formula was used:

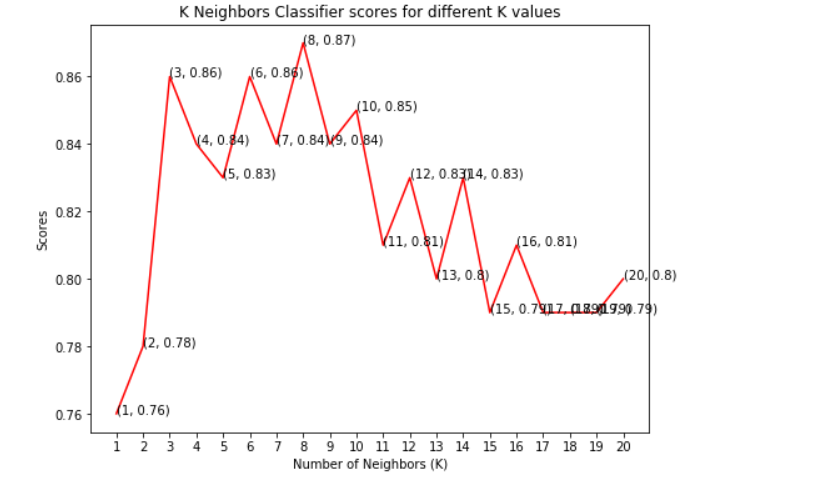


**Experiments/ Results / Discussion:**

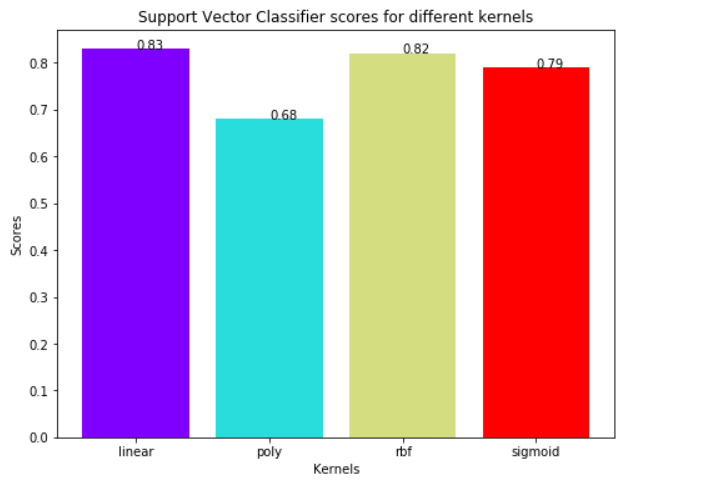
In order to solve these two classification models, several models were tested. For heart disease prediction, we used the following models and tested them according to their accuracies.

* K Neighbors Classifier
* Support Vector Classifier
* Decision tree Classifier
* Random forest Classifier

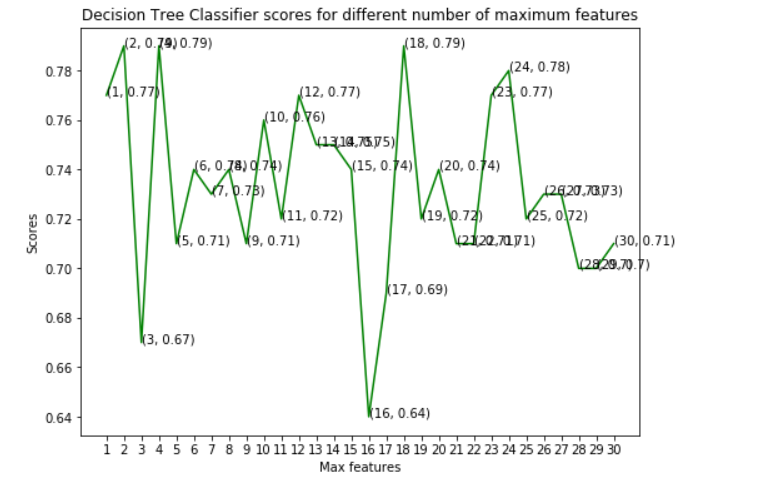
First we tested these models individually, for Knn, in order to find the most suitable values of K, we tested the model on different values of K.



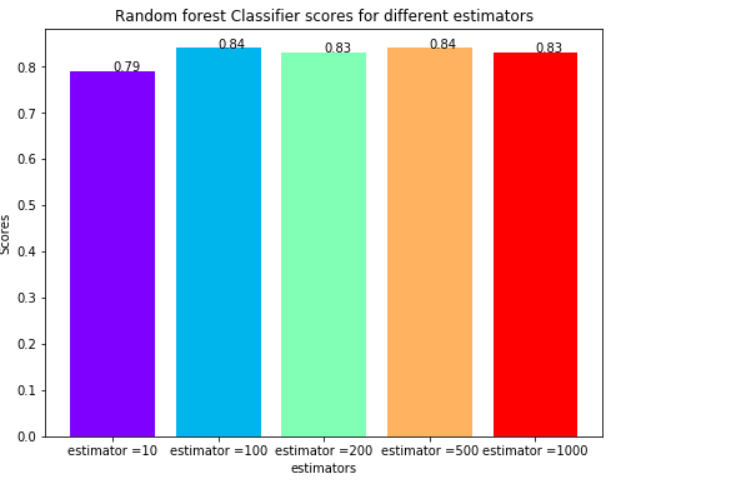
Here we can see that K = 8 gives the highest accuracy i.e. 87. Next we tested Support Vector classifiers on different Kernels, the results are as shown



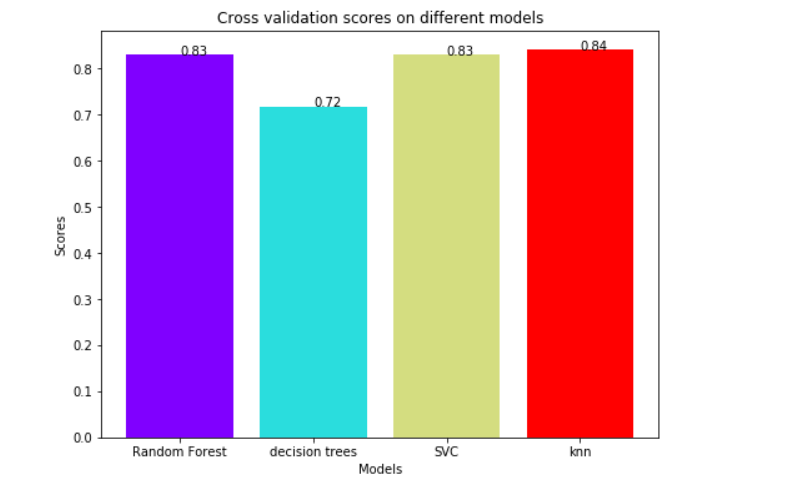
Here the linear Kernel performed the best with an accuracy of 83%. Next we tested using Decision tree classifier with different max\_features parameter.



Here we got the maximum accuracy of 79% using max features 2, 4 and 18. Next we tested random forest classifier using different values of n\_estimators.



Here we got the best accuracy of 84% with estimators 100 and 500. Finally after finding the most suitable parameters of these models, we tested them using 10 – fold cross validation in order to select our final model.

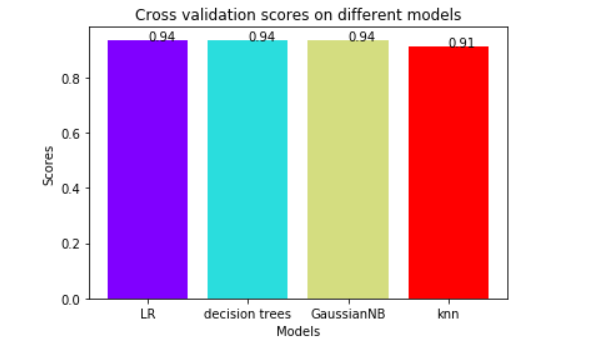


On cross validation K-NN is giving the best accuracy i.e. 84% and on test data, KNN for k = 8 was giving accuracy of 87%, therefore we selected KNN as our model for predicting heard diseases.

For disease prediction, we tested the accuracy the following models.

1. Decision Tree
2. Knn
3. Gaussian Naive Bayes
4. Logistic Regression

For disease prediction we directly applied cross validation to test the accuracy of these models on default parameters.

****

Here we can see that three algorithms are giving the same accuracy i.e. 94%. However we used decision trees with default parameters.

**Conclusion:**

This project was made to ease the communication b/w doctor and patient and to use machine learning for helping doctors to diagnose patients. This report has focused on the machine learning aspect of this project. We tried many different models that are used for solving classification problems and tested them individually on test data and finalized them using cross validation. KNN and decision trees were selected on the basis of their cross validation accuracies. In the future a lot of work can done to extend and to improve the functionalities in this project. We can collect more data and do more data engineering to better understand the underlying trend in data and train better models. We can also use deep learning, which can give us better accuracies on these problems. There are a lot off applications out there for machine learning in health care, and in the future we will try to explore more problems and integrate them in our application, in order to give more resources to the doctor and improving the health care industry.

**References:**

[1] <https://towardsdatascience.com/heart-disease-prediction-73468d630cfc>

[2] Akash C. Jamgade, Prof. S. D. Zade, “Disease Prediction Using Machine Learning”, International Research Journal of Engineering and Technology (IRJET), May.2019

**Datasets:**

Heart Disease dataset: <https://www.kaggle.com/ronitf/heart-disease-uci>

Disease prediction from symptoms dataset: <https://www.kaggle.com/neelima98/disease-prediction-using-machine-learning>