Prediction of Chronic Kidney Disease Using Machine Learning Algorithm

**1.1Introduction:**

Early detection of preventable diseases is important for better disease management and more efficient health-care resource allocation.Our project focuses on to ease the medical processes of detecting the chronic kidney disease by providing a platform or a website where the users can enter their medical values to predict their disease.

*Machine Learning* is a growing field concerned with the study of enormous and several variable data having computational methods, algorithms and techniques for analysis and prediction. Machine learning delivers methodoligies ,approaches that can help resolving analytic and predictive In Medical Science’s viewpoint, Machine Learning techniques have showed success in prediction and diagnosis of numerous critical diseases.

Due to complexity and vagueness of data engendered by healthcare transactions, it is impossible to analyze them with traditional tools. In order to make the decision-making process easier and more trustable, machine learning techniques are provided to transmute these data into useful information and makes it feasible to get useful results and patterns and trends out of these huge amounts of data.

Machine Learning has been widely used in many areas. One of these areas which is using it even more and more as an essential tool is healthcare management. All agents in a healthcare industry can significantly benefit Data mining applications. Different machine learning classification algorithm for diagnosis of chronic kidney disease is discussed.

Various classification techniques that are used are: Decision Tree, KNN Classifier,Support Vector Classifier,Random Forest Classifier,Logistic Regression,NaiveBayes Classifier.

We have used Python version-3 for our project purpose.Python is an interpreted , high-level, general-purpose programming language.It can be used to handle big data and perform complex mathematics. *Python community* has developed many modules to help programmers implement machine learning.  It has syntax that allows developers to write programs with fewer lines than some other programming languages.

We have even built a web UI that uses tools such as **Node**-**RED** and IBM **Watson** Studio.Usingthe **machine learning service** of IBM cloud with features for training and deploying the machine learning models,we have intgegrated and deployed our model.Next,We have built our own Web UI using Node -Red of WatsonStudio.

**1.2Objective:**

Chronic Kidney Disease prediction is one of the most central problems in medical decision making because it is one of the leading cause of death.

The aim of this work is *to reduce the diagnosis time* and to improve the diagnosis accuracy through classification algorithms . Machine Learning techniques have showed success in prediction and diagnosis of numerous critical diseases.So,We tried to provide a user interface that uses our ML model so that an individual just by sitting at their homes could know if his/her symptoms would lead to Kidney disease or not.

Our project model analyzes the symptoms provided by the user as input and gives the probability of the disease as an output. *Disease Prediction* is done by implementing the **Logistic Regression** Algorithm.

We have also developed a web UI based on our algorithm helps in prediction of chronic kidney disease, providing a user friendly interface and making it more approachable to individuals.

**1.3Problem Statement:**

In Medical Science’s viewpoint, Machine Learning techniques have showed success in prediction and diagnosis of numerous critical diseases. Chronic Kidney Disease prediction is one of the most central problems in medical decision making because it is one of the leading cause of death.

We, through this project, are trying to take the advantage of available technologies to develop prediction models for Chronic Kidney disease.We have compared the performance of six classifiers in the prediction of chronic kidney disease.

We have even got an accuracy of 1.0 with Random Forest Model but what we understood is that it is overfitting (Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data).Other than that, The experimental results of our proposed method have demonstrated that logistic regression has produced superior prediction performance in terms of classification accuracy(96.67%).

Also,we have built a Web tool to make the prediction approachable to individuals and also the diagnosis time of the disease can be reduced.

2.Review of Literature

* DSVGK Kaladhar, Krishna Apparao Rayavarapu and Varahalarao Vadlapudi et al applied machine learning techniques to predict kidney stones by using C4.5, Random forest (93%), Support Vector Machines (SVM), Logistic, NN and Naive Bayes machine learning algorithms which becomes useful in automating the treatment of kidney stones diseases
* Vijayarani, S., Dhayanand, M. S., & Phil, M. (2015). Kidney disease prediction using svm and ann
* S. Ramya and Dr. N. Radha, “Diagnosis of Chronic Kidney Disease Using Machine Learning Algorithms”, International Journal of Innovative Research in Computer and Communication Engineering, Volume 4, Issue 1, January 2016, pp 813- 820
* SA, S. (2013). Intelligent heart disease prediction system using data mining techniques. International Journal

of Healthcare & Biomedical Research, 1, 94-101.

* Chapman, P., Clinton, J., Kerber, R. Khabeza, T., Reinartz, T., Shearer, C., Wirth, R.: “CRISP-DM 1.0: Step by step data mining guide”, SPSS, 1-78, 2000.

3. **Data Collection**

For study and analysis, I used the dataset from the UCI Machine Learning Repository named Chronic Kidney Disease. Total 400 instances of the dataset is used for the training to prediction algorithms, out of which 250 has label chronic kidney disease (CKD) and 150 has label non chronic kidney disease (NCKD)

Features for chronic kidney disease used in experiment:

|  |  |  |
| --- | --- | --- |
| **S.No** | **Attribute** | **Description** |
| 1. | Age(numerical) | Age in years |
| 2. | Blood Pressure(numerical) | bp in mm/Hg |
| 3. | Specific Gravity(nominal) | sg-(1.005,1.010,1.015,1.020,1.025) |
| 4. | Albumin(nominal) | al-(0,1,2,3,4,5) |
| 5. | Sugar(nominal) | Su-(0,1,2,3,4,5) |
| 6. | Red Blood Cells(nominal) | rbc-(normal,abnormal) |
| 7. | Pus Cell(nominal) | pc-(normal,abnormal) |
| 8. | Pus Cell Clumps(nominal) | pcc-(present,notpresent) |
| 9. | Bacteria(nominal) | ba-(present,notpresent) |
| 10. | Blood Glucose Random(numerical) | bgr in mgs/dl |
| 11. | Blood Urea (numerical) | bu in mgs/dl |
| 12. | Serum Creatinine (numerical) | sc in mgs/dl |
| 13. | Sodium (numerical) | sod in mEq/L |
| 14. | Potassium (numerical) | pot in mEq/L |
| 15. | Haemoglobin (numerical) | hemo in gms |
| 16. | Packed Cell Volume (numerical) | Pcv |
| 17. | White Blood Cell Count (numerical) | wc in cells/cumm |
| 18. | Red Blood Cell Count (numerical) | rc in millions/cmm |
| 19. | Hypertension (nominal) | htn - (yes, no) |
| 20. | Diabetes Mellitus (nominal) | dm - (yes, no) |
| 21. | Coronary Artery Disease (nominal) | cad - (yes, no) |
| 22. | Appetite (nominal) | appet - (good, poor) |
| 23. | Pedal Edema (nominal) | pe - (yes, no) |
| 24. | Anemia (nominal) | ane - (yes, no) |
| 25. | Class (nominal) | class - (ckd, notckd) |

There are total twenty-four features, most of which are clinical in nature and the rest are physiological. As a part of data pre-processing, missing values and outliers are imputed with mean value of that feature for continuous data and attribute model value for categorical data. Nominal data are converted to numerical values. For example, Nominal are labelled “1” and ‘Abnormal’ are labelled “0”.The training data that is present generally has missing attributes and uneven data. Hence it is required to fill the missing values by taking the mean of the other values used in the data set.

**4. Methodology:**

values ‘Normal’ are labelled “1” and ‘Abnormal’ are labelled “0”.

This section describes the proposed methodology for data mining from CKD dataset. The very first and important step is preprocessing and cleaning the data.

**Cleaning process** is the process of filling the missing values based on either there are categorical or nominal data. This process also involves replacing unknown values (like '?','\t43' as we have in our data).Also,we identified three numerical columns of our data to be in string format.So , we tried convert it into numerical format as we can only process the numerical data.

For identifying the columns with null values,we plotted a heat map (fig1)

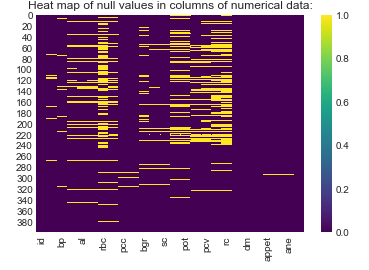


fig1.heat map of null values in dataset

Here the yellow plots(1.0) indicate presence of null values.

In columns of categorical data,we replaced with mode value

data['rbc'].fillna('normal',inplace=True)

data['pc'].fillna('normal',inplace=True)

data['pcc'].fillna('notpresent',inplace=True)

data['ba'].fillna('notpresent',inplace=True)

data['htn'].fillna('no',inplace=True)

data['dm'].fillna('no',inplace=True)

data['cad'].fillna('no',inplace=True)

data['appet'].fillna('good',inplace=True)

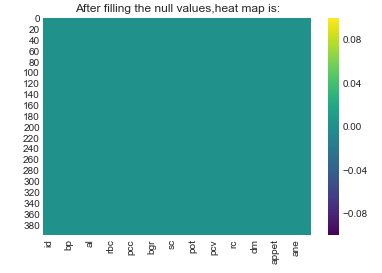
data['pe'].fillna('no',inplace=True)

data['ane'].fillna('no',inplace=True)

In columns of numerical data,we replaced the null values with mean values.

data.fillna(data.mean(),inplace=True)

After replacing null values,heat map would be like:(indicating that the all null values are removed(0.0).

**fig2:**.heat map of null values in dataset

Now , We have to encode the categorical data.For example:

enc= {"rbc":{"normal": 1, "abnormal": 0},

"pcc": {"present": 1, "notpresent": 0},

"htn": {"yes": 1, "no": 0},

"dm": {"yes": 1, "no": 0,"\tno":0,"\tyes":1," yes":1},

"cad": {"yes": 1, "no": 0,"\tno":0},

"appet": {"good": 1, "poor": 0},

"ane": {"yes": 1, "no": 0},

"classification":{"ckd":1,"notckd":0,"ckd\t":1}}

data.replace(enc, inplace=True)

**4.1.1.Plottings for understanding and visualising the data:**

For better understanding of influence of the attributes on the target variable 'classification', , let's take a look at the relationships between numeric features and other numeric features by considering the correlation heat map(fig3):

\***Correlation** is a value between -1 and 1 that represents how closely values for two separate features move in unison.

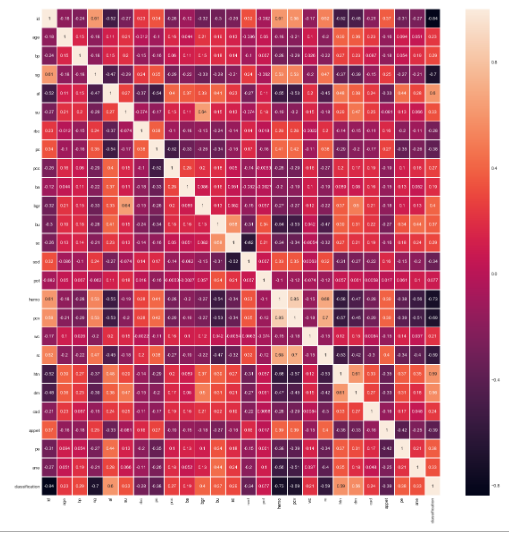
\*Positive correlation means that as one feature increases, the other increases; eg. a child's age and her height.

\*Negative correlation means that as one feature increases, the other decreases; eg. hours spent studying and number of parties attended.

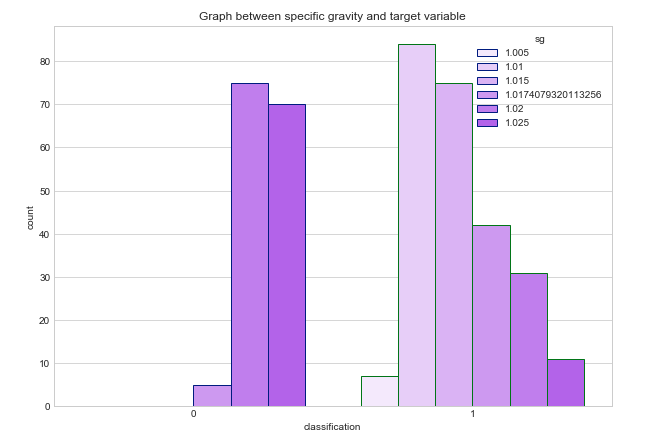
\*Correlations near -1 or 1 indicate a strong relationship.

\*Those closer to 0 indicate a weak relationship.

\*0 indicates no relationship.

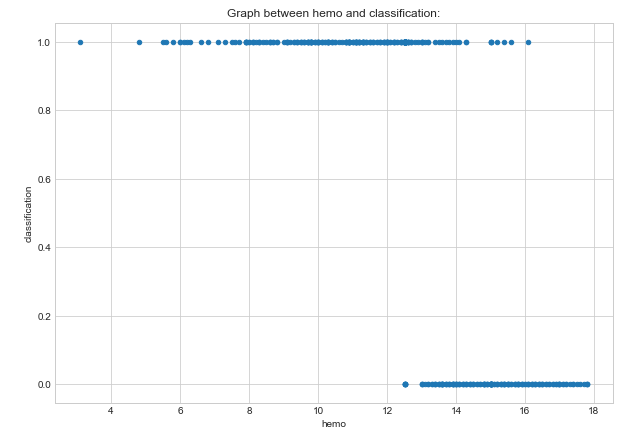


**fig3.**Correlation graph using heat map

Let us plot the attributes that influence more(like hemo,sg,su..)

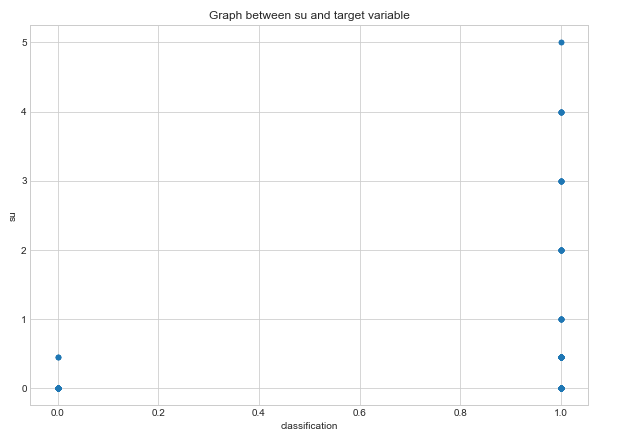
**fig4.**countplotbetween sugar 'sg' and target variable 'classification'

Here , in fig4,We can observe that the maximum values of 'sg' (specific gravity) less than1.02 are influencing the classification(target variable) to be 1(ckd).The Values of sg equal to 1.02 and 1.025 are more influencing the target variable to be 0(not ckd).



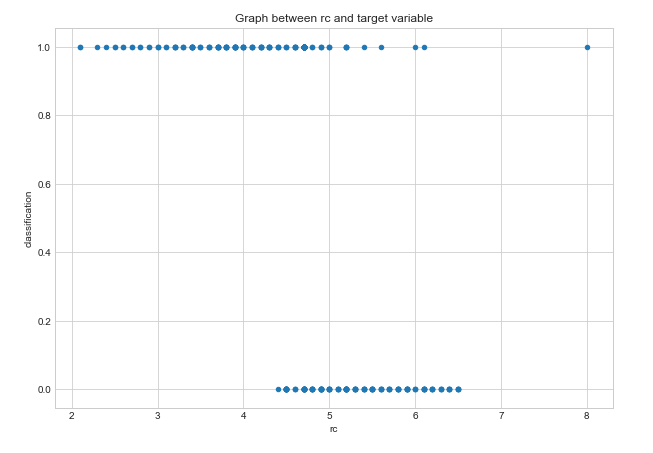
**fig5.** scatter plot between 'hemo' and target variable 'classification'

Here, in fig5,the plottings of 'hemo' (hemoglobin) make a sigmoid shaped curve , as the 'hemo' increases from 12 they are influencing the target variable to be 0.0(notckd).



**fig6.**scatterplot between sugar 'su' and target variable 'classification'

Here,in fig6,We can observe that as the 'su' values increase (from su=1)the target variable is more proned to be 1.0(ckd).



**fig7.** sctterplot between 'rc' and target variable 'classification'

Here, in fig5,the plottings of 'rc' (hemoglobin) make a sigmoid shaped curve , as the 'hemo' increases from rc=4 they are influencing the target variable to be 0.0(notckd).

From fig3. We can observe that the attribute 'pot'(potassium) is having a correlation of 0.077 which is closer to 0(zero).Let us understand its influence on output with visualisation using graph.

We can observe (from fig8) that the influence of 'pot' on target variable is not clear.We can see that some of the values less than 10 are in classification=1.0 and some are in classification=0.0.

**4.2Data Modelling:**

We know that our target variable 'classification' is a binary variable(categorical variable).For prediction task of chronic kidney disease we separately applied six machine learning algorithms namely: like Support Vector Classifier,NaiveBayes,KNN classifier,Logistic Regression,Decision Tree Classifier,Random Forest Classifier.

**Data Preparation:**

Dividing the given data into input array and output(target) variable.Then using train\_test\_split method to split the data into training data(70%) and testing data(30%).

#x is input features and y is target variable

from sklearn.cross\_validation import train\_test\_split

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=1)

**Model-1 Logistic Regression:**

First as we observed a sigmoid curve in the plottings,we started with Logistic Regression. Logistic regression is basically a supervised classification algorithm. Logistic regression models the data using the sigmoid function.

**Python Implementation :**

from sklearn.linear\_model import LogisticRegression

lr=LogisticRegression()

lr.fit(X\_train,Y\_train)

prediction=lr.predict(X\_test)

from sklearn.metrics import accuracy\_score

accuracy\_score(Y\_test,prediction)

accuracy \_score has been given as 96.667

**Model-2 KNN Classifier:**

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

**Python Implementation :**

from sklearn.neighbors import KNeighborsClassifier

kfold = model\_selection.KFold(n\_splits=10, random\_state=7)

model\_k = KNeighborsClassifier()

results = model\_selection.cross\_val\_score(model\_k,x,y, cv=kfold)

print(results.mean()\*100)

It gives us the accuracy of model\_k to be 69.5

**Model-3 Random Forest Classifier:**

Python implementation:

from sklearn.ensemble import RandomForestClassifier

random\_forest = randomForestClassifier(n\_estimators=12,max\_features=3)

random\_forest.fit(X\_train, Y\_train)

Y\_prediction = random\_forest.predict(X\_test)

print(accuracy\_score(Y\_test,Y\_prediction)\*100)

It gives us an accuracy of 100%

**Model-4 NaiveBayes Classifier:**

 It is a classification technique based on **Bayes**' Theorem with an assumption of independence among predictors. In simple terms, a **Naive Bayes** classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Python implementation:

from sklearn.naive\_bayes import GaussianNB

gnb = GaussianNB()

gnb.fit(X\_train, Y\_train)

NY\_predict=gnb.predict(X\_test)

print(accuracy\_score(Y\_test,NY\_predict)\*100)

It gives us an accuracy of 95.83%

**Model-5 DecisionTree Classifier:**

**Decision Trees** are a type of Supervised **Machine Learning** (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter.

Python implementation:

from sklearn.tree import DecisionTreeClassifier

model\_d = DecisionTreeClassifier()

model\_d.fit(X\_train,Y\_train)

Y\_pred=model\_d.predict(X\_test)

accuracy\_score(Y\_pred,Y\_test)

It gives us an accuracy of 95.83%

**Model-6 Support Vector Classifier:**

A **Support Vector Classifier** (**SVC**) is a discriminative **classifier** formally defined by a separating hyperplane.Support vector machine algorithm finds a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

Python implementation:

from sklearn.svm import SVC

from sklearn import model\_selection

kfold = model\_selection.KFold(n\_splits=10, random\_state=7)

model\_s = SVC()

results = model\_selection.cross\_val\_score(model\_s, x, y, cv=kfold)

print(results.mean()\*100)

It gives us an accuracy of 62.5%.

Out of all these , we finally used the Logistic regression model , as it has a better accuracy of 96.67%.

Let us find out the accuracy of Logistic regression manually using the confusion matrix:

from sklearn.metrics import confusion\_matrix

confusion\_matrix(Y\_test,prediction)

Output:

array([[49, 1],

[ 3, 67]], dtype=int64)

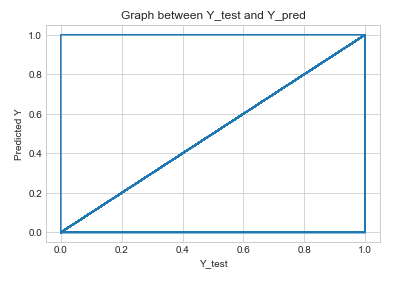
Now,the accuracy is :

(49+67)/(49+1+3+67)=0.9667

Error is :

(1+3)/(49+1+3+67)=0.0333

Now let us plot the graph between Y\_test and Y\_predict of logistic regression model (fig9.)



**fig9.** plot between Y\_test and predicted\_y

**Findings :**

Table of accuracy of different classifier models used:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Model** | **Accuracy** | **Error** |
| 1. | Random Forest | 1.000 | 0.0 |
| 2. | Logistic Regression | 0.967 | 0.033 |
| 3. | NaiveBayes | 0.958 | 0.042 |
| 4. | DecisionTree | 0.958 | 0.042 |
| 5. | KNN classifier | 0.695 | 0.305 |
| 6. | SupportVectorClassifier | 0.625 | 0.375 |

The highest accuracy is obtained in case of Random Forest Classifier with 100% accuracy .It seems to fit the training dataset,but has poor fit with new datasets,i.e, Over fitting . ***Overfitting*** happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

After Random Classifier the highest accurate model is Logistic Regression with **96.67%.**Also the plottings of attributes effect on the output seems to be a sigmoid curve.So finally it is observed that the **Logistic Regression** model is better model for Chronic Kidney Disease Prediction.

**Limitations and Suggestions:**

1.The size of the dataset used in this research is still quite small. A large dataset would definitely give better results.

2.Also,If we work on even more large sets and try to reduce the overfitting problem with random forest , we can have a more accurate prediction model.

**Conclusion**:

Accurate prediction of chronic kidney disease is one of the emerging topics in medical diagnosis. Early detection of preventable diseases is important for better disease management.So, with the objective to find the better model for chronic kidney disease, We have compared the performance of six classifiers in the prediction of chronic kidney disease. The experimental results of our proposed method have demonstrated that Logistic Regression has produced superior prediction performance in terms of classification accuracy for our considered dataset, with accuracy 96.67% . Thus finally it is observed thatLogistic Regression is better algorithm for chronic kidney diagnosis

**Web based application:**

We incorporated a responsive web based frontend for better and easier experience for users.They could easily enter the values of the parameters and can see the predicted output .

The web based tool is made using Watson Studio's Node Red as shown in fig10.

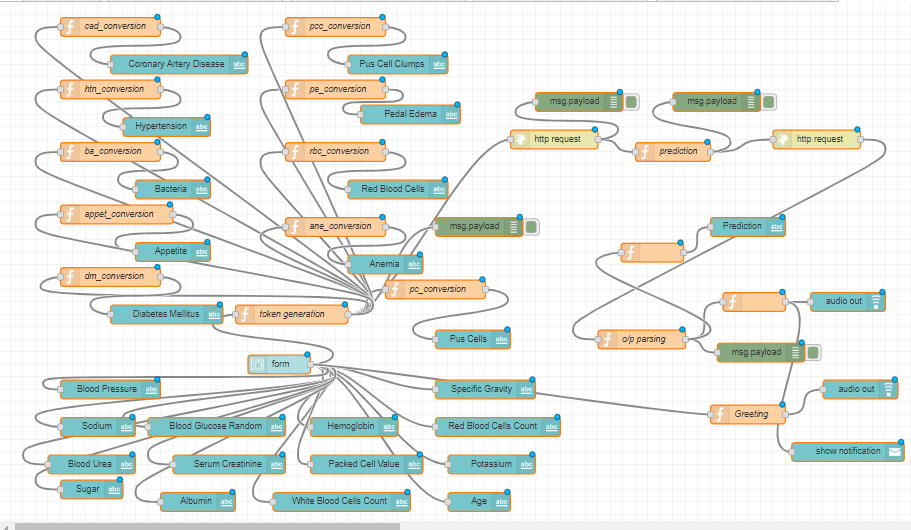
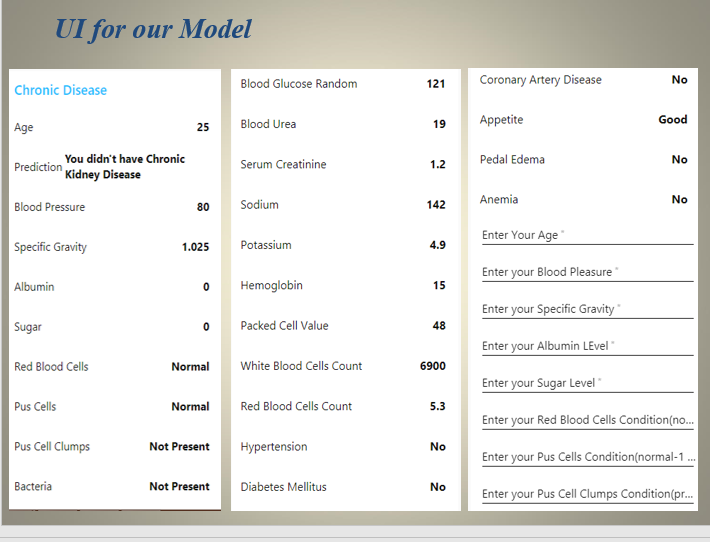


fig10.Node-Red model of the web based UI

Our model is integrated to it using the Machine learning Service .The UI is highly responsive .The UI of prediction is fig 11.



**Future Work:**

The experimental results of our proposed method have demonstrated that RandomForest has produced superior prediction performance in terms of classification accuracy,but may be overifitted for our data set. It was also observed that few classifiers have yielded poor classification accuracy. This problem will be investigated in our future study by

(i) Exploring all possible combination of various different types of input features and different machine learning techniques.

(ii) By dealing with various factors that affects prediction performance , for improving the prediction accuracy and finally identifying the exact cause and improve the diagnosis accuracy through classification algorithms.

In future,I am also planning to develop a web tool which will be helpful in prediction of all the huge chronic diseases , which reduces the diagnosis time of diseases.