IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

Prediction Model for Detection of Chronic Kidney Disease: Logistic Regression

Vishesh S[1], Pavan Kumar C K[2], Varnitha N[3]

BE, Department of TCE, VTU, Belgaum, India[1]

BE, Department of CSE, Adichunchanagiri Institute of Technology, India[2]

BE, Department of ECE, RNSIT, Bangalore, India[3]

Abstract: Urinary System in human beings consists of two bean shaped elements called Kidneys. Purpose of urinary system is to eliminate waste, control levels of electrolytes, regulate blood pressure and blood volume, and control metabolism. Any malfunctioning of the elements of the urinary system may lead to imbalance of other connected systems of the body like circulatory system, digestive system and nervous system. Chronic Kidney Disease (CKD) is one ailment which could devastate the human body. It can be prevented via examining few indicators like RBC count, specific gravity value, Blood Pressure (BP), albumin levels in urine, sugar content, anaemia and WBC count. Other conditions like coronary artery disease, Diabetes Mellitus (DM) and bacterial infections could directly affect the kidneys. In this paper we have collected 400 samples from a public hospital and selected fields have been analysed for designing a prediction model for CKD. Logistic regression is carried out and accuracy, precision, and f1 score of the model has been measured. Various conclusions can be drawn from this interdependent data set and can be stored as historical data for future analysis.

Keywords: Urinary System in human beings, Chronic Kidney Disease (CKD), RBC count, specific gravity value, Blood Pressure (BP), albumin levels in urine, sugar content, anaemia, WBC count, Logistic regression, accuracy, precision, and f1 score, coronary artery disease, Diabetes Mellitus (DM) and bacterial infections

I. INTRODUCTION

Chronic Kidney Disease is fatal in human beings if left untreated and undetected. We have identified several factors contributing to the failure of kidneys. Few of the listed fields are

i. Age
ii. Blood Pressure (BP)
iii. Specific gravity
iv. Albumin levels in urine
v. Diabetes Mellitus
vi. RBC count, WBC count, pus cell and packed cell volume
vii. Presence or absence of hypertension, coronary artery disease and pedal edema

II. PROBLEM STATEMENT

400 patients were subjected to various tests and the resulting data was recorded in .csv format. The tabulated results need to be analysed statistically and data visualisation [1] to be carried out on the data set. The data set may contain missing values and data pre-processing needs to be carried out on missing values redundant data and non-numerical values. Logistic regression to create a prediction model for detecting Chronic Kidney Disease for real-time samples.

```
In [2]:

import pandas as pd
import numpy as np
import seaborn as sns
%matplotlib inline
import matplotlib.pyplot as plt
```

Figure 1 shows the Python code to import libraries.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

III. METHODOLOGY

A. Importing Libraries [2]: Figure 1 shows the Python code to import libraries. We have used three libraries

- numpy is a package for scientific computing with Python. This library is imported as 'np' and will be used throughout the project.
- pandas is for data manipulation and analysis. panadas is an open source, BSD- licenced library providing easy-to-use data structures and data analysis tools. pandas is imported as pd.
- matplotlib.pyplot is a collection of command style functions that make matplotlib work like MATLAB. It is imported as plt
- seaborn is a Python data visualization library based on matplotlib for attractive and informative statistical graphics.

B. Importing data: Figure 2 shows the Python code to import data from respective directory/ file and assigning it to DataFrame df. The data stored in CSV format is being imported. [3] [4]

C. Checking for NaN: It is very essential in data pre-processing to check for NaN. Figure 3 shows the Python code to check for NaN. In this attempt we could identify few NaN.

D. Manipulating NaN values: It is essential to remove the NaN values. This can be done by

- Removing the entire column containing many NaN values
- Forward fillna method
- Backward fillna method
- Mean method

Figure 4 shows the technique of forward fillna method and figure 5 shows the method of dropping the column.

E. Plotting a Heatmap: Correlation between the fields of the recorded data is analysed by plotting a heatmap. The values may be negative or positive and the magnitude plays a key role in designing various predictive models in AI. Figure 6 shows a heatmap and correlation model.

F. Splitting the data into train and test sets. Figure 7 shows the python code to split the data set into train and test data.

G. Applying logistic regression on the split data. Figure 8 shows logistic regression on given data set.

```
import matplotlib.pyplot as plt
df = pd.read_csv('kidney.csv')
```

Figure 2 shows the Python code to import data and assigning it to DataFrame df

```
In [6]:
df.isnull().sum()
Out[6]:
id                      0
age                     9
bloodpressure          12
specificgravity        47
albumin                46
sugar                  49
redbloodcells         152
puscell                65
puscellclumps           4
bacteria                4
bloodglucoserandom     44
bloodurea              19
serumcreatinine        17
sodium                 87
potassium              88
haemoglobin            52
packedcellvolume       71
whitebloodcellcount   105
redbloodcellcount     130
hypertension            2
diabetesmellitus        2
coronaryarterydisease   2
appetite                1
pedaledema              1
anemia                  1
classification          0
dtype: int64
```

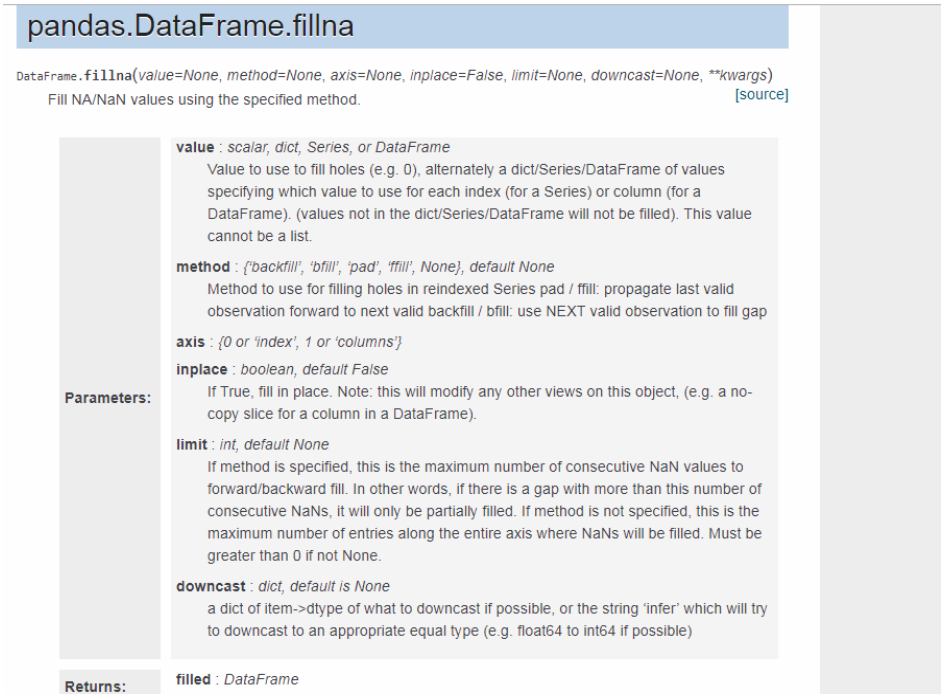
Figure 3 shows the Python code to check for NaN.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

In [8]:

```
df.drop(["redbloodcells","whitebloodcellcount","redbloodcellcount"],axis=1,inplace=True)
```

Figure 5 shows the method of dropping the column



pandas.DataFrame.fillna

DataFrame.**fillna**(*value=None, method=None, axis=None, inplace=False, limit=None, downcast=None, **kwargs*)

Fill NA/NaN values using the specified method. [source]

Parameters:

value : *scalar, dict, Series, or DataFrame*
Value to use to fill holes (e.g. 0), alternately a dict/Series/DataFrame of values specifying which value to use for each index (for a Series) or column (for a DataFrame). (values not in the dict/Series/DataFrame will not be filled). This value cannot be a list.

method : *{'backfill', 'bfill', 'pad', 'ffill', None}, default None*
Method to use for filling holes in reindexed Series pad / ffill: propagate last valid observation forward to next valid backfill / bfill: use NEXT valid observation to fill gap

axis : *{0 or 'index', 1 or 'columns'}*

inplace : *boolean, default False*
If True, fill in place. Note: this will modify any other views on this object, (e.g. a no-copy slice for a column in a DataFrame).

limit : *int, default None*
If method is specified, this is the maximum number of consecutive NaN values to forward/backward fill. In other words, if there is a gap with more than this number of consecutive NaNs, it will only be partially filled. If method is not specified, this is the maximum number of entries along the entire axis where NaNs will be filled. Must be greater than 0 if not None.

downcast : *dict, default is None*
a dict of item->dtype of what to downcast if possible, or the string 'infer' which will try to downcast to an appropriate equal type (e.g. float64 to int64 if possible)

Returns:

filled : *DataFrame*

Figure 4 shows the technique of forward fillna method

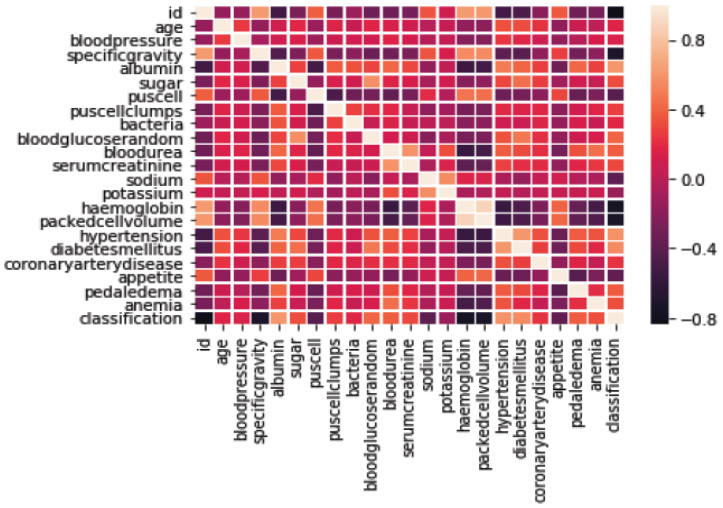


Figure 6 shows a heatmap and correlation model.

In [20]:

```
from sklearn.model_selection import train_test_split
```

In [77]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=12)
```

Figure 7 shows the python code to split the data set into train and test data.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 4, April 2019

```
In [82]:
from sklearn.linear_model import LogisticRegression

In [85]:
logmodel= LogisticRegression()
logmodel.fit(X_train,y_train)
```

Figure 8 shows logistic regression on given data set.

IV. RESULTS

Figure 7 shows the plot of hypertension, Diabetes Mellitus v/s CKD respectively. Figure 10 shows the results of logistic regression model. Figure 11 shows the Accuracy score of the designed model. From this data, precision, f1 score and reliability can be calculated.

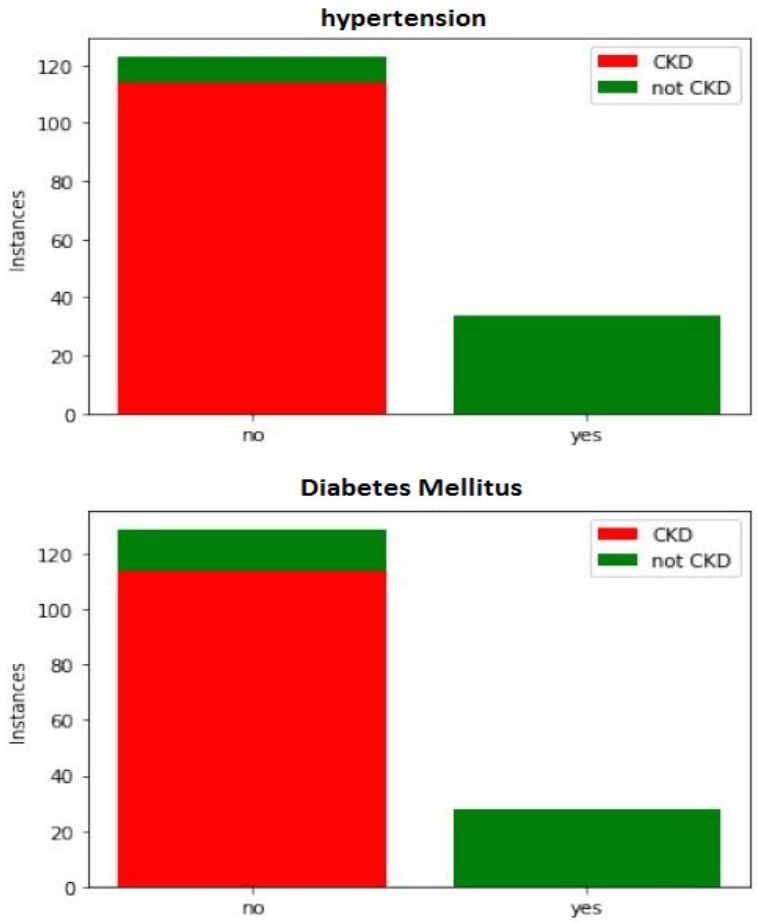


Figure 9 shows the plot of hypertension, Diabetes Mellitus v/s CKD

```
Out[85]:

LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False)
```

Figure 10 shows the results of logistic regression model

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 4, April 2019

In [86]:

```
predictions= logmodel.predict(X_test)
predictions
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,predictions)
from sklearn.metrics import accuracy_score
accuracy_score(y_test,predictions)
```

Out[86]:

```
0.9833333333333333
```

Figure 11 shows the Accuracy score of the designed model.

V. CONCLUSION

Chronic Kidney Disease is fatal, but can be treated and cured when identified at an early stage. 400 samples were considered to design a predictive model using Logistic Regression. The data set was taken from a trusted source, pre-processed, statistically analysed and graphs plotted. A heatmap was plotted to identify the correlation between different fields of interest. The data being cleansed (removing NaN values) was subjected to division as train and test data. 70% of the data was fed for training and the remaining considered for test. We have calculated the accuracy of the model and were happy to conclude with 98.33% accuracy. Any new samples taken can be predicted with this model with high reliability, accuracy and precision.

REFERENCES

[1]. Data Analytics: Car Sales in a Calendar Year Vishesh S1 , Pavan Kumar C K2- IJARCCE, vol 8, issue 4, April 2019.
[2]. [2]. The Python Standard Library — Python 3.7.1rc2 documentation https://docs.python.org/3/library/
[3]. Data Warehousing Architecture and Pre-Processing- Vishesh S, Manu Srinath, Akshatha C Kumar, Nandan A.S.- IJARCCE, vol 6, issue 5, May 2017.
[4]. Data Mining and Analytics: A Proactive Model - http://www.ijarcce.com/upload/2017/february-17/IJARCCE%20117.pdf

BIOGRAPHY

VISHESH S born on 13th June 1992, hails from Bangalore (Karnataka) and has completed B.E in Telecommunication Engineering from VTU, Belgaum, Karnataka in 2015. He also worked as an intern under Dr. Shivananju BN, former Research Scholar, Department of Instrumentation, IISc, Bangalore. His research interests include Embedded Systems, Wireless Communication, BAN and Medical Electronics. He is also the Founder and Managing Director of the corporate company Konigtronics Private Limited. He has guided over a hundred students/interns/professionals in their research work and projects. He is also the co-author of many International Research Papers. He is currently pursuing his MBA in e-Business and PG Diploma in International Business. Presently Konigtronics Private Limited has extended its services in the field of Software Engineering and Webpage Designing. Konigtronics also conducts technical and non-technical workshops on various topics.