A MACHINE LEARNING APPROCH FOR PREDICTING CHRONIC KIDNEY DISEASE

[1]C. JAYALAXMI, [2]CHENNA [3]AKSHAY KUMAR, [4]MOHAMMED MUQTAR, [5]SANGEM SAI SOURABH, [6]VARDHAN SOMARAM

Dept. of Information Technology, TKR College of Engineering and Technology, Hyderabad, Telangana

jayalakshmi@tkrcet.com, chenna.akshay123@gmail.com, muqtar635@gmail.com, kingsourabhsangam@gmail.com, vardhanvardhan12@gmail.com.

ABSTRACT:

Chronic kidney disease (CKD) is a global health problem with high morbidity and mortality rate, and it induces other diseases. Since there are no obvious symptoms during the early stages of CKD, patients often fail to notice the disease. Early detection of CKD enables patients to receive timely treatment to ameliorate the progression of this disease. Machine learning models can effectively aid clinicians achieve this goal due to their fast and accurate recognition performance. In this study, we propose a machine learning methodology for diagnosing CKD. The CKD dataset was obtained from the University of California Irvine (UCI) machine learning repository, which has a large number of missing values. KNN imputation was used to fill in the missing values, which selects several complete samples with the most similar measurements to process the missing data for each incomplete sample. Missing values are usually seen in real-life medical situations because patients may miss some measurements for various reasons. After effectively filling out the incomplete data set, six machine learning algorithms (logistic regression, random forest, support vector machine-nearest neighbor, naïve Bayes classier and feed forward neural network) were used to establish models. Among these machine learning models, random forest achieved the best performance with 99.75% diagnosis accuracy. By analyzing the misjudgments generated by the established models, we proposed an integrated model that uses random forest algorithm which could achieve an average accuracy of 99.83% after ten times of simulation. Hence, we speculated that this methodology could be applicable to more complicated clinical data for disease diagnosis.

1. INTRODUCTION

CHRONIC kidney disease (CKD) is a global public health problem affecting approximately 10% of the world's population. The percentage of prevalence of CKD in China is 10.8% and the range of prevalence is 10%-15% in the United States. According to another study, this percentage has reached 14.7% in the Mexican adult general population. This disease is characterized by a slow deterioration in renal function, which eventually causes a complete loss of renal function. CKD does not show obvious symptoms in its early stages. Therefore, the disease may not be detected until the kidney loses about 25% of its function. In addition, CKD has high morbidity and mortality, with a global impact on the human body. It can induce the occurrence of cardiovascular disease.

CKD is a progressive and irreversible pathologic syndrome. Hence, the prediction and diagnosis of CKD in its early stages is quite essential, it may be able to enable patients to receive timely treatment to ameliorate the progression of the disease.

Machine learning refers to a computer program, which calculates and deduces the information related to the task and obtains the characteristics of the corresponding pattern. This technology can achieve accurate and economical diagnosis of diseases. Hence, it might be a promising method for diagnosing CKD. It has become a new kind of medical tool with the development of information technology and has a broad application prospect because of the rapid development of electronic health record. In the medical field, machine learning has already been used to detect human body status analyze the relevant factors of the disease and diagnose various diseases.

2. LITERATURE SURVEY:

Kunwar, et al. entitled Analysis is Using Data Mining Classification Techniques" published in 2016. Data mining is the process of extracting hidden information from massive dataset, categorizing valid and unique patterns in data. There are many data mining techniques like clustering, classification, association analysis, regression etc. The objective of the paper is to predict (CKD) using classification techniques like Naive Bayes and Artificial Neural Network (ANN). The experimental results implemented in Rapid Miner tool show that Naive Bayes produce more accurate results.

Amirgaliyev, et al. entitled "Analysis Dataset by Applying Machine Learning Methods" published in 2015. Currently, there are many people in the world suffering from chronic kidney diseases worldwide. Due to the several risk factors like food, environment and living standards many people get diseases suddenly without understanding of their condition. In this research study, the effects of using clinical features to classify patients with chronic kidney disease by using support vector machines algorithm is investigated. The chronic kidney disease data set is based on clinical history, physical examinations, and laboratory tests *Devika, et al.* entitled "Comparative Study of Classifier for Prediction Using Naive Bayes, KNN and Random Forest" published in 2019. Chronic

Kidney disease defines constrains which affects your kidneys and reduces your potential to stay healthy. Machine learning is an important task as it benefits many applications, varied knowledge mining classification approaches and machine learning algorithms are applied for prediction of chronic diseases. Therefore, this paper examines the performance of Naive Bayes, K-Nearest Neighbor (KNN) and Random Forest classifier on the basis of its accuracy, preciseness and execution time for CKD prediction. Finally, the outcome after conducted research is that the performance of Random Forest classifier is finest than *Naive Bayes and KN Avci E et al*. entitled "Performance Comparison of Some Classifiers on Data" published in 2018. In this study, dataset named obtained from UCI database is used. The dataset consists of 400 individual's information and contains 25 features dataset was classified according to whether it is chronic kidney disease using Naive Bayes (NB), K-Star, Support Machines (SVM) and J48 classifiers used in data mining *Dulhare, et al*. entitled "Extraction of Action Rules for using Naive Bayes Classifier" published in 2017. Chronic kidney disease (CKD), also known as chronic renal disease, which is a progressive loss in kidney function over a period of months or years. It is defined by the presence of kidney damage or decreased glomerular filtration rate (GFR). The estimated prevalence of CKD is about 9-13 % in the general adult population is a silent condition. Signs and symptoms of CKD, if present, are generally not specific in nature and unlike several other chronic diseases (such as congestive heart failure and chronic obstructive lung disease), they do not reveal a clue for diagnosis or severity of the condition. Early detection and treatment can often keep chronic kidney disease from getting worse *Aljaaf, et al*. entitled "Early Prediction of Using Machine Learning Supported by Predictive Analytics" published in 2018.Chronic Kidney Disease is a serious lifelong condition that induced by either kidney pathology or reduced kidney functions. Early prediction and proper treatments can possibly stop, or slow the progression of this chronic disease to end-stage, where dialysis or kidney transplantation is the only way to save patient's life.

3. PROPOSED SYSTEM

To summarize the previous CKD diagnostic models, we find that most of them suffering from either the method used to impute missing values has a limited application range or relatively low accuracy. Therefore, in this work, we propose a methodology to extend application range of the CKD diagnostic models. At the same time, the accuracy of the model is further improved. The contributions of the proposed work are as follows.

- We used KNN imputation to fill in the missing values in the data set, which could be applied to the data set with the diagnostic categories are unknown.
- Previously, Logistic regression (LOG), Random Forest, SVM, KNN, naive Bayes classifier (NB) and feed forward neural network (FNN) were used to establish CKD diagnostic models on the complete CKD data sets. The models with better performance were extracted for misjudgment analysis.
- An integrated model that uses Random Forest algorithm was established and it improved the performance of the component models in CKD diagnosis after the missing values were filled by KNN imputation. KNN imputation is used to fill in the missing values. To our knowledge, this is the first time that KNN imputation has been used for the diagnosis of CKD. In addition, building an integrated model is also a good way to improve the performance of separate individual models. The proposed methodology might effectively deal with the scene where patients are missing certain measurements before being diagnosed. In addition, the resulting integrated model shows a higher accuracy. Therefore, it is speculated that this methodology might be applicable to the clinical data in the actual medical diagnosis.

Modules

Feature-Extraction

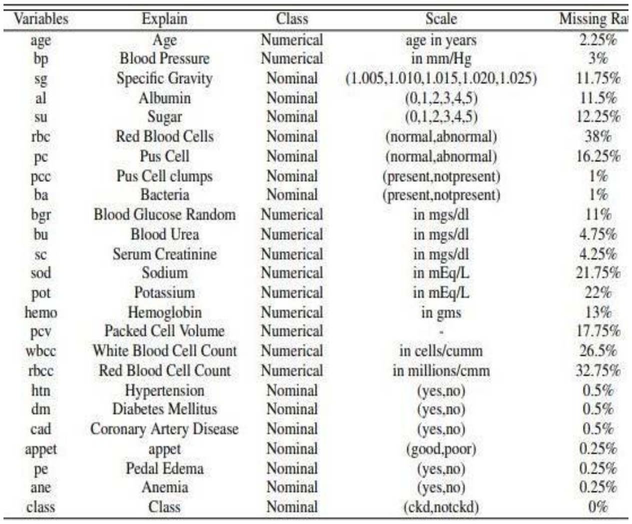
This section involves the convolutionary layers that obtain image features from the resize images and is also joined after each convolution with the ReLU. Max and average pooling of the feature extraction decreases the size. Ultimately, both the convolutional and the pooling layers act as purifiers to generate those image characteristics.

Classification

The final step is to classify images, to train deep learning models along with the labeled images to be trained on how to recognize and classify images according to learned visual patterns. The authors used an open-source implementation via the TensorFlow module, using Python and OpenCV including the VGG-16 CNN model.

Dataset Description

The CKD data set used in this study was obtained from the UCI machine learning repository [32], which was collected from hospital and donated by *Soundara pandian et al*. on 3 rd July, 2015. The data set contains 400 samples. In this CKD data set, each sample has 24 predictive variables or features (11 numerical variables and 13 categorical (nominal) variables) and a categorical response variable (class). Each class has two values, namely, ckd (sample with CKD) and notckd (sample without CKD). In the 400 samples, 250 samples belong to the category of ckd, whereas 150 samples belong to the category of notckd. It is worth mentioning that there is a large number of missing values in the data.



Variables	Explain	Class	Scale	Missing Ra
age	Age	Numerical	age in years	2.25%
bp	Blood Pressure	Numerical	in mm/Hg	3%
sg	Specific Gravity	Nominal	(1.005,1.010,1.015,1.020,1.025)	11.75%
al	Albumin	Nominal	(0,1,2,3,4,5)	11.5%
su	Sugar	Nominal	(0,1,2,3,4,5)	12.25%
rbc	Red Blood Cells	Nominal	(normal,abnormal)	38%
pc	Pus Cell	Nominal	(normal,abnormal)	16.25%
pcc	Pus Cell clumps	Nominal	(present,notpresent)	1%
ba	Bacteria	Nominal	(present,notpresent)	1%
bgr	Blood Glucose Random	Numerical	in mgs/dl	11%
bu	Blood Urea	Numerical	in mgs/dl	4.75%
sc	Serum Creatinine	Numerical	in mgs/dl	4.25%
sod	Sodium	Numerical	in mEq/L	21.75%
pot	Potassium	Numerical	in mEq/L	22%
hemo	Hemoglobin	Numerical	in gms	13%
pcv	Packed Cell Volume	Numerical	-	17.75%
wbcc	White Blood Cell Count	Numerical	in cells/cumm	26.5%
rbcc	Red Blood Cell Count	Numerical	in millions/cmm	32.75%
htn	Hypertension	Nominal	(yes,no)	0.5%
dm	Diabetes Mellitus	Nominal	(yes,no)	0.5%
cad	Coronary Artery Disease	Nominal	(yes,no)	0.5%
appet	appet	Nominal	(good,poor)	0.25%
pe	Pedal Edema	Nominal	(yes,no)	0.25%
ane	Anemia	Nominal	(yes,no)	0.25%
class	Class	Nominal	(ckd,notckd)	0%

DATA PROCESSING

Each categorical (nominal) variable was coded to facilitate the processing in a computer. For the values of rbc and pc, normal and abnormal were coded as 1 and 0, respectively. For the values of pcc and ba, present and notpresent were coded as 1 and 0, respectively. For the values of htn, dm, cad, pe and ane, yes and no were coded as 1 and 0, respectively. For the value of appet, good and poor were coded as 1 and 0, respectively. Although the original data description defines three variables sg, al and su as categorical types, the values of these three variables are still numeric based, thus these variables were treated as numeric variables. All the categorical variables were transformed into factors. Each sample was given an independent number that ranged from 1 to 400. There is a large number of missing values in the data set, and the number of complete instances is 158. In general, the patients might miss some measurements for various reasons before making a diagnosis. Thus, missing values will appear in the data when the diagnostic categories of samples are unknown, and a corresponding imputation method is needed. After encoding the categorical variables, the missing values in the original CKD data set were processed and filled at first. KNN imputation was used in this study, and it selects the K complete samples with the shortest Euclidean distance for each sample with missing values. For the numerical variables, the missing values are filled using the median of the corresponding variable in K complete samples, and for the category variables, the missing values are filled using the category that has the highest frequency in the corresponding variable in K complete samples. For physiological measurements, people with similar physical conditions should have similar physiological measurements, which is the reason for using the method based on a KNN to fill in the missing values. For example, the physiological measurements should be stable within a certain range for healthy individuals. For diseased individuals, the physiological measurements of the person with a similar degree of the same disease should be similar. In particular, the differences in physiological measurements data should not be large for people with similar situations. This method should also be adapted to the diagnostic data of other diseases, as it has been applied in the area of hyperuricemia

When the median of corresponding variables in K complete samples are selected, K is preferably taken as an odd number because in this case the middle number is naturally the median when the values of the numeric variables in the K complete samples are sorted by numerical value. The selection of K should neither be too large nor too small. An excessively large K value may ignore the inconspicuous mode, which might be important. Conversely, an excessively small K value causes noise and the abnormal data affects the filling of the missing values exceedingly. Therefore, the values of K in this work were chosen as 3, 5, 7, 9 and 11. As a result, five complete CKD data sets were generated.

One is to use random values to fill in the missing values; the other is to use mean and mode of the corresponding variables to fill in missing values of continuous and categorical variables, respectively

Implementation

Framing the results from our provider interviews utilizing NPT enabled us to identify key barriers and critical junctures where interventions need to occur to address these barriers Additionally, NPT can guide the choice of interventions likely to be most effective, whether they are cognitive vs. tangible or practice vs. process etc. For example, the lack of coherence around CKD may best be addressed by academic mentoring from clinical experts in CKD, while the lack of reflexive monitoring might be addressed by providing practices with data management systems and personnel.

The TRANSLATE CKD trial currently underway is utilizing some of these strategies in a multi-faceted intervention to address some of these barriers across each of the NPT constructs. For example, academic mentors meet monthly with the primary care clinicians to discuss clinical questions related to CKD and reinforce the guidelines (coherence and cognitive participation), while a data team pulls and compiles practice-level performance data to assist practices in monitoring their progress (reflexive monitoring), and practice facilitators engage with practices in quality improvement projects to improve workflows and processes around CKD (cognitive participation and collective action).

In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like PYTHON 3.6, Anaconda Cloud are used for coding. With respect to the type of application, the right programming language is chosen.

Testing

In this phase the system is tested. Normally programs are written as a series of individual modules, this subject to separate and detailed test. The system is then tested as a whole. The separate modules are brought together and tested as a complete system. The system is tested to ensure that interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of data (volume testing) and that the system does what the user requires (acceptance/beta testing).

Maintenance

Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

Jupyter notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more**.**

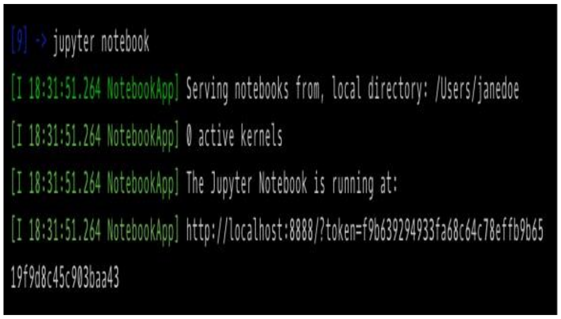
Jupyter Notebooks are a powerful way to write and iterate on your Python code for data analysis. Rather than writing and re-writing an entire program, you can write lines of code and run them one at a time. Then, if you need to make a change, you can go back and make your edit and rerun the program again, all in the same window.

Jupyter Notebook is built off of *IPython*, an interactive way of running Python code in the terminal using the *REPL model* (Read-Eval-Print-Loop). The IPython Kernel runs the computations and communicates with the Jupyter Notebook front-end interface. It also allows Jupyter Notebook to support multiple languages. Jupyter Notebooks extend IPython through additional features, like storing your code and output and allowing you to keep markdown notes.

If you'd rather watch a video instead of read an article, please watch the following instructions on how to use a Jupyter Notebook. They cover the same information.

Launch A Notebook

To launch a Jupyter notebook, open your terminal and navigate to the directory where you would like to save your notebook. Then type the command jupyter notebook and the program will instantiate a local server at localhost:8888 (or another specified port).

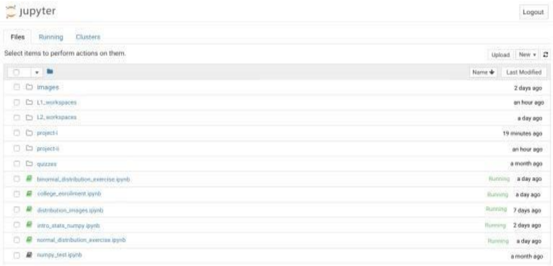


```
[9] -> jupyter notebook
[I 18:31:51.264 NotebookApp] Serving notebooks from, local directory: /Users/janedoe
[I 18:31:51.264 NotebookApp] 0 active kernels
[I 18:31:51.264 NotebookApp] The Jupyter Notebook is running at:
[I 18:31:51.264 NotebookApp] http://localhost:8888/?token=f9b639294933fa68c64c78effb9b65
19f9d8c45c903baa43
```

A browser window should immediately pop up with the Jupyter Notebook interface, otherwise, you can use the address it gives you. The notebooks have a unique token since the software uses pre-built Docker containers to put notebooks on their own unique path. To stop the server and shutdown the kernel from the terminal, hit control-C twice.

Jupyter Interface

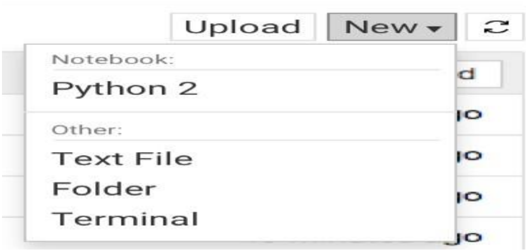
Now you're in the Jupyter Notebook interface, and you can see all of the files in your current directory. All Jupyter Notebooks are identifiable by the **notebook icon** next to their name. If you already have a Jupyter Notebook in your current directory that you want to view, find it in your files list and click it to open.



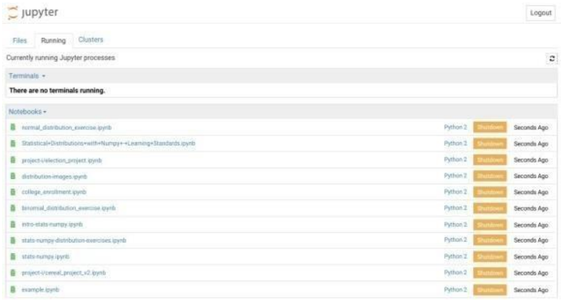
Select items to perform actions on them.

Name		Last Modified
images		2 days ago
L1_workspaces		an hour ago
L2_workspaces		a day ago
project-i		19 minutes ago
project-ii		an hour ago
quizzes		a month ago
binomial_distribution_exercise.ipynb	Running	a day ago
college_enrollment.ipynb	Running	a day ago
distribution_images.ipynb	Running	7 days ago
intro_stats_numpy.ipynb	Running	2 days ago
normal_distribution_exercise.ipynb	Running	a day ago
numpy_test.ipynb		a month ago

To create a new notebook, go to **New** and select **Notebook - Python 2**. If you have other Jupyter Notebooks on your system that you want to use, you can click **Upload** and navigate to that particular file.



Notebooks currently running will have a green icon, while non-running ones will be grey. To find all currently running notebooks, click on the **Running** tab to see a list.



jupyter   Logout

Files | Running | Clusters

Currently running Jupyter processes

Terminals ▾

There are no terminals running.

Notebooks ▾

normal_distribution_exercise.ipynb	Python 2	Shutdown	Seconds Ago
Statistical+Distributions+with+Numpy+-+Learning+Standards.ipynb	Python 2	Shutdown	Seconds Ago
project-i/election_project.ipynb	Python 2	Shutdown	Seconds Ago
distribution-images.ipynb	Python 2	Shutdown	Seconds Ago
college_enrollment.ipynb	Python 2	Shutdown	Seconds Ago
binomial_distribution_exercise.ipynb	Python 2	Shutdown	Seconds Ago
intro-stats-numpy.ipynb	Python 2	Shutdown	Seconds Ago
stats-numpy-distribution-exercises.ipynb	Python 2	Shutdown	Seconds Ago
stats-numpy.ipynb	Python 2	Shutdown	Seconds Ago
project-i/cereal_project_v2.ipynb	Python 2	Shutdown	Seconds Ago
example.ipynb	Python 2	Shutdown	Seconds Ago

Inside The Notebook

When you open a new Jupyter notebook, you'll notice that it contains a *cell*.

Cells are how notebooks are structured and are the areas where you write your code. To run a piece of code, click on the cell to select it, then press SHIFT+ENTER or press the play button in the toolbar above.

Additionally, the **Cell** dropdown menu has several options to run cells, including running one cell at a time or to run all cells at once.

4. SYSTEM DESIGN

System design is transition from a user-oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

Software Design Requirements

In designing the software following principles are followed:

i. ***Modularity and partitioning***: software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.

ii. ***Coupling:*** modules should have little dependence on other modules of a system.

iii. ***Cohesion:*** modules should carry out in a single processing function.

iv. ***Shared use*:** avoid duplication by allowing a single module be called by other that need the function it provide

Architectural Design

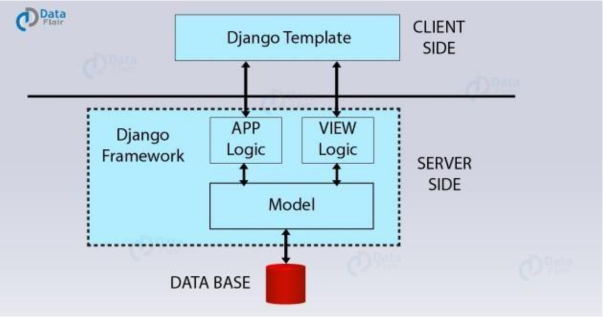
Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application.

MVT Structure has the following three parts:

Model: Model is going to act as the interface of your data. It is responsible for maintaining data. It is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).

View: The View is the user interface — what you see in your browser when you render a website. It is represented by HTML/CSS/JavaScript and Jinja files.

Template: A template consists of static parts of the desired HTML output as well as some special syntax describing how dynamic content will be inserted.



Technical Architecture

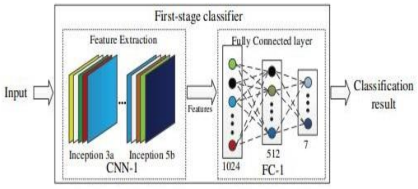


Fig. 2. The architecture of the first-stage classifier.

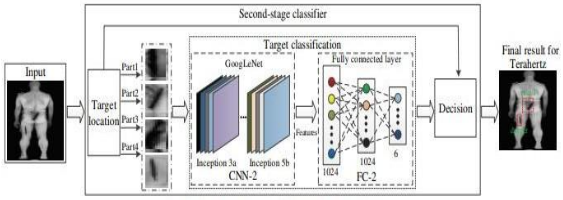
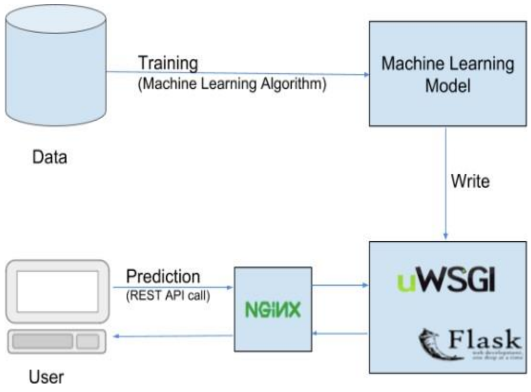


Fig. 3. The architecture of the second-stage classifier.

There may be more steps involved, depending on what specific requirements you have, but below are some of the main steps:



Test cases

No	Test name	Inputs	Expected output	Actual Output	status

1	Load Dataset	Csv file	Read dataset	Load dataset	success
2	Split dataset	Train80% and test20%	Divide the training set and Testing set	Split train and Test	success
	Train Model	Train dataset, random value, predicted class	Train with best accuracy	Train with best accuracy	success
4	Validate Model	No .of Epochs	Validate the Model with best fit	Model Generated	success
5	Predict accuracy and Error Rate	Accuracy	Plot expected accuracy and predicted accuracy	Plot expected predicted accuracy	success
6	Test Data	Test column	Predicted accuracy	Predicted accuracy	success

5. CONCLUSION:

The proposed CKD diagnostic methodology is feasible in terms of data imputation and samples diagnosis. After unsupervised imputation of missing values in the data set by using KNN imputation, the integrated model could achieve a satisfactory accuracy. Hence, we speculate that applying this methodology to the practical diagnosis of CKD would achieve a desirable effect. In addition, this methodology might be applicable to the clinical data of the other diseases in actual medical diagnosis.

However, in the process of establishing the model, due to the limitations of the conditions, the available data samples are relatively small, including only 400 samples. Therefore, the generalization performance of the model might be limited. In addition, due to there are only two categories (ckd and notckd) of data samples in the data set, the model cannot diagnose the severity of CKD. In the future, a large number of more complex and representative data will be collected to train the model to improve the generalization performance while enabling it to detect the severity of the disease. We believe that this model will be more and more perfect by the increase of size and quality of the data.

REFERENCES:

1.Z.Chenetal.,"Diagnosis of patients with chronic kidney disease by using twofuzzyclassifiers,"Chemometr.Intell.Lab .,vol.153,pp.140-145,Apr. 2016.

2.A. Subasi, E. Alickovic, J. Kevric, "Diagnosis of chronic kidney disease by using random forest," in Proc. Int. Conf. Medical and Biological Engineering, Mar. 2017, pp. 589-594.

3.L. Zhang et al., "Prevalence of chronic kidney disease in china: a crosssectional survey," Lancet, vol. 379, pp. 815-822, Aug. 2012.

4.A. Singhetal. "Incorporating temporal HER data in predictive models for risk stratification of renal function deterioration," J. Biomed. Inform., vol. 53, pp. 220-228, Feb. 2015.

5.A. M. Cueto-Manzano et al., "Prevalence of chronic kidney disease in an adultpopulation,"Arch.Med.Res.,vol.45,no .6,pp.507-513,Aug.2014.

6.H.Polat,H.D.Mehr,A.Cetin,"Diagnosis of chronic kidney disease based on support vector machine by feature selection methods," J. Med. Syst., vol. 41, no. 4, Apr. 2017.

7.C. Barbieri et al., "A new machine learning approach for predicting the response to anemia treatment in a large cohort of end stage renal disease patient sunder going dialysis,"Comput. Biol. Med.,vol.61,pp.56-61,Jun. 2015.

8.V. Papademetriou et al., "chronic kidney disease, basal insulin glargine, and health outcomes in people with dysglycemia: The origin study," Am. J. Med., vol. 130, no. 12, Dec. 2017.

9.N. R. Hill et al., "Global prevalence of chronic kidney disease - A systematic review and meta-analysis," Plos One, vol. 11, no. 7, Jul. 2016.

10. M. M. Hossainetal., "Mechanical anisotropy assessment in kidney cortex using ARFI peak displacement: Preclinical validation and pilot in vivo