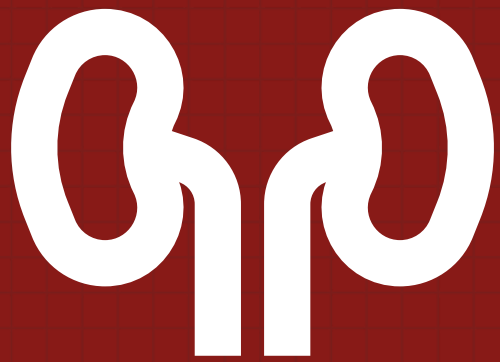
Proposed Title:

Chronic Kindey disease Risk prediction



Presented by Group 50:

Harsh Kumar Sharma
20205037

Rishav Raj
20205122

Sathpati Deepak
20205142

Supervised by: Dr. Satish Chandra

Electronics and Communication Engineering Department

Abstract

This study will highlight the importance of combining machine learning with expert knowledge when trying to predict Chronic Kidney Disease (CKD). To develops a web application with user interface to prompt out whether a patient is prone to CKD or not. To necessitate this, the prominent features will be selected using feature selection techniques and classification will be preformed by widely used classifiers. Each feature will be computed by a score using feature Selection methods. Different Classification Algorithms will be used to predict the class labels. The model will be trained with data set and using K Fold Cross validation method the dataset will be partitioned into training and testing data. It will try to achieve its maximum accuracy.

Introduction

What is CKD?

Chronic kidney disease (CKD) means your kidneys are damaged and can't filter blood the way they should. The disease is called "chronic" because the damage to your kidneys happens slowly over a long period of time. This damage can cause wastes to build up in your body.

Its causes:

The two main causes of CKD are diabetes and high blood pressure.
Diabetes is elevated blood sugar levels, harming the blood vessels, eyes, kidneys, and heart.
Ineffective management of high blood pressure can contribute significantly to heart attack, stroke, and chronic renal illness. [1]

What is need of predicting early CKD?

Because a person may only survive without their kidneys for an average of 18 days, dialysis and kidney transplants are in great demand. It is crucial to possess reliable techniques for CKD early prediction.

Literature Reviews

Many researchers are working on the prediction of CKD, and these researchers are receiving the appreciated results from their models.

Author	Journal	Year	Title	Methodology Used	Result
Mohd. Elhoseny, et. al. [2]	Scientifc Reports	2019	Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease	D-ACO algorithm	D-ACO framework performs FS, ACO based learning and removes irrelevant features. Using dataset, the efficiency is evaluated, and a comparison is also made with the existing methods.
Arezoo Haratian, et. al. [3]	Scientific Reports	2022	Detection of factors affecting kidney function using machine learning methods	Multiple Imputation	Relationship between Vitamin D and blood creatinine levels can be a factor effecting kidney functioning
Qiong Bai, Chunyan Su, et. al. [4]	Scientific Reports	2022	Machine learning to predict end stage kidney disease in chronic kidney disease	Naïve Bayes, random forest, decision tree, and K-nearest neighbors	This study showed the feasibility of ML in evaluating the prognosis of CKD. Logistic regression, naïve Bayes and random forest demonstrated comparable predictability to the KFRE in this study.

Author	Journal	Year	Title	Methodology Used	Result
Koyner, Jay L, et. al. [5]	Clinical investiga-tions	2020	The development of a machine learning inpatient acute kidney injury prediction model	Gradient boosting machine	Electronic health record data can be used to predict impending acute kidney injury prior to changes in serum creatinine with excellent accuracy
Catalina M., Gabriela, et. al. [6]	Clinical research	2020	Renal function decline in latinos with type 2 diabetes	Retrospective study	Uncontrolled high blood pressure, overweight, and longstanding t2dm duration at baseline were significantly associated with increased risk of ckd
Kate, Rohit J., Ruth M., et. al. [7]	BMC Medical Informatics and Decision Making	2016	Prediction and detection models for acute kidney injury in hospitalized older adults	logistic regression, support vector machines, decision trees and naïve Bayes	Logistic regression performed the best for AKI detection (AUC 0.743) and was a close second to the ensemble for AKI prediction
Akbilgic O., Praveen K., et. al. [8]	Clinical research	2022	Machine learning to identify dialysis patients at high death risk	Random forest	mean age of our cohort was 68.7 ± 11.2 years, 98.1% of patients were men and 71.4% were diabetic

Objectives

1

To develop model with higher accuracy using the identified features and classifier combination.

2

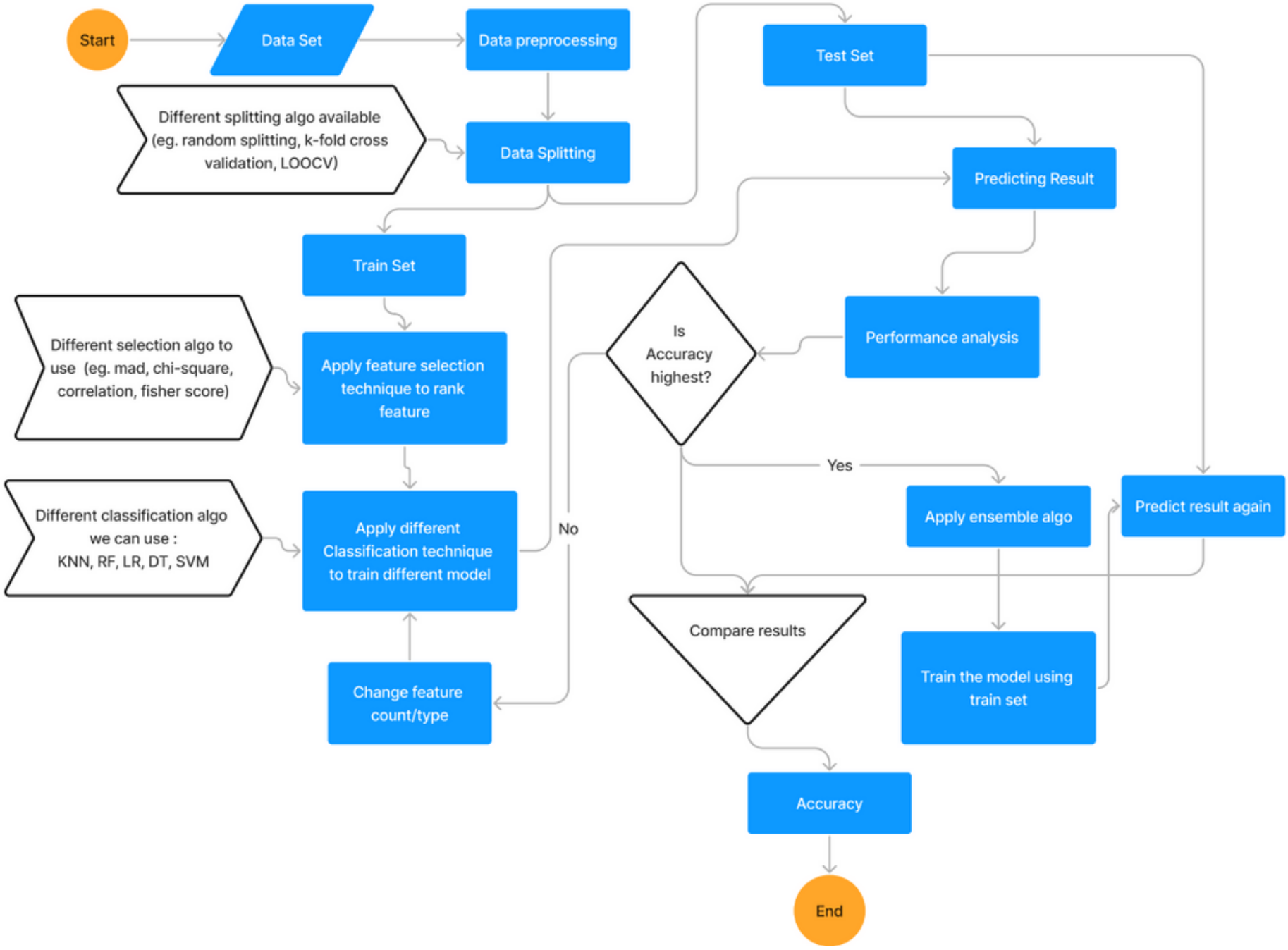
To develop a web application for diagnosing the risk factor of kidney.

Methodology

The project will use the dataset provided by UCI [9] repository for training and testing the model. It will first pre-process the dataset to remove redundant and faulty data, then split the dataset to train and test data using data dividing algorithm. Then train data will be used to do feature selection by using different algorithm and comparing the result's accuracy, to rank the features according to it. This will help in finding the best set of features and classifier. It will then use ensemble algorithm to combine different classifier to increase the accuracy of final model.

The project will be done in Jupyter Notebook with the help of python as the programming language and sklearn as the library for using different Machine Learning algorithms. The web page will be made using React.

Flow Diagram



Road map

Literature Survey	Doing literature Survey to analyze the preexisting model and come up with a better model
Datasets	To collect the real time data from the UCI repository of patients and do data pre-processing.
Feature Selection	To determine the parameters that affect the trend of chronic kidney disease using the rank method
Optimal Classification	To train the model using different classification methods (KNN, Random Forest, decision tree) and determine the best among them.
Final model and user interface	To determine the evaluation functions (accuracy, precision, and recall) of the model using classification and feature selection and further using ensemble model to predict the result. Finally, creating a web page for user to interact with.

Work done so far

1. **We have done many literature reviews and collected information about previous works in this field.**

2. **We have done the pre processing of data that was available on the University of California Irvine(UCI) website.**

3. **We have used few feature selection methods like correlation, mean absolute difference to find the top features available in the datasets.**

4. **We have used different classifier methods and different numbers of top features to build models and see there results.**

5. **We have made the user interface using web technology like react and tailwind.**

Project snippets

Database loading and pre-processing

[3]:
```
#Database loading and pre-processing

from sklearn.model_selection import train_test_split
from sklearn.impute import KNNImputer
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler

#Create function for checking missing values which accepts a dataframe as its parameter
def null_values_check(df):
    #Error handling to prevent abnormal termination of operation
    try:
        #if-else statement for null value check
        if(df.isnull().values.any() == True):
            #if there are null values present, print a column-wise summary of records with null values
            print('Number of null records within each column:\n' + str(df.isnull().sum()))
        else:
            print('There is no missing values in the dataset.')

    except Exception as e:
        logging.error(e)

#initialise variable with dataset name
dataset_name = '.\chronic_kidney_disease.csv'

#error-handling to prevent abnormal termination of code
try:
    #import and load weather dataset into pandas dataframe
    chronic_kidney_disease_dataframe = pd.read_csv(dataset_name)
```

Converting all the data types to float

[12]:
```
# taking the length of the column and converting the each column to float
colLength = (len(chronic_kidney_disease_dataframe.axes[1]))

# for storing the position of each column
count=0;

for i in chronic_kidney_disease_dataframe:
    count += 1
    # the last column of the data set is label so when we reach last column just break the loop
    if(count==25):
        break

    # converting each column to float data type

    chronic_kidney_disease_dataframe[i] = chronic_kidney_disease_dataframe[i].astype('float64')
```

handling null values ¶

[16]:
```
chronic_kidney_disease_dataframe.columns
```

[16]:
```
Index(['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
       'sc', 'sod', 'pot', 'hemo', 'pcv', 'wbcc', 'rbcc', 'htn', 'dm', 'cad',
       'appet', 'pe', 'ane', 'class'],
      dtype='object')
```

[17]:
```
features = ['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu',
           'sc', 'sod', 'pot', 'hemo', 'pcv', 'wbcc', 'rbcc', 'htn', 'dm', 'cad',
           'appet', 'pe', 'ane']
```

Replacing the null value with the median

[18]:
```
for feature in features:
    chronic_kidney_disease_dataframe[feature] = chronic_kidney_disease_dataframe[feature].fillna(chronic_kidney_disease_dataframe[feature].median())
```

[19]:
```
chronic_kidney_disease_dataframe.isnull().any().sum()
```

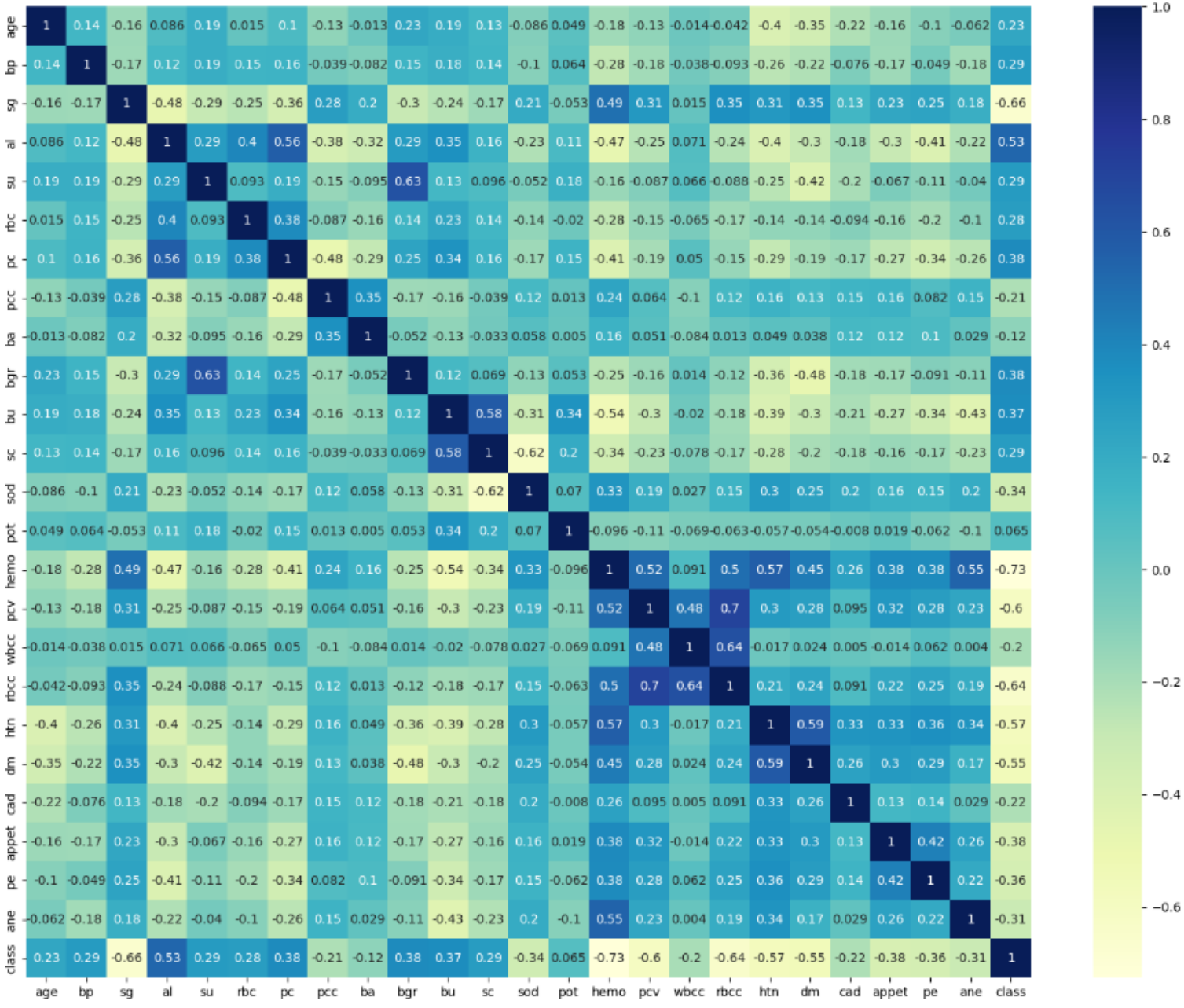
[19]:
```
0
```

Function for calling different classifier and selecting the best feature from the correlation

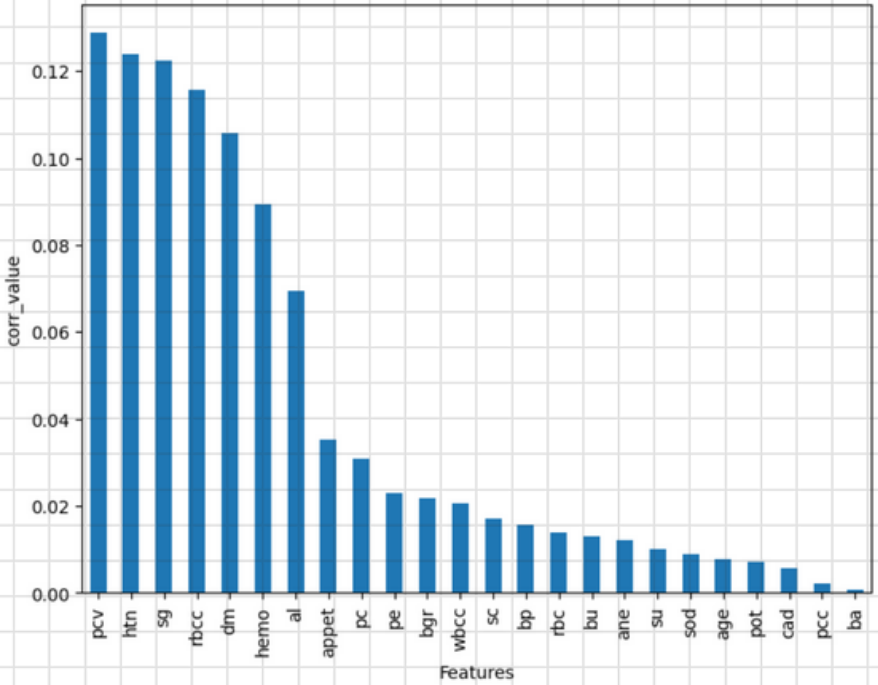
[27]:

```
# selecting the top feature
def selectFeature(n):
    return ranked_features.nlargest(n).index

# for selecting all the details of the dataset based on the selected top feature
# chronic_kidney_disease_dataframe[['sg', 'htn', 'hemo', 'dm', 'al', 'appet', 'rc', 'pc']]
dataset=[]
def topFeature(feature):
    dataset=[]
    topFeature = []
    for i in feature:
        topFeature.append(i)
    dataset.append(topFeature)
    return dataset


# Importing Performance Metrics:
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# the classifier function

# calling classifier
def classifierUsed(selectedDataSetForTraining,feature,number):
    randomForestClassifier(selectedDataSetForTraining,feature,number)
    decisionTreeClassifier(selectedDataSetForTraining,feature,number)
    KNNClassifier(selectedDataSetForTraining,feature,number)
    logisticRegressionClassifier(selectedDataSetForTraining,feature,number)
    supportVectorMachineClassifier(selectedDataSetForTraining,feature,number)


# function for random forest classifiers
def numberOfFeatureForClassifier(number):
    selectedFeature = selectFeature(number)
    feature =[]
    for i in selectedFeature:
        # appending the selected feature into a list
        feature.append(i)

    selectedDataSetForTraining = topFeature(feature)

    # calling the classifier function
    classifierUsed(selectedDataSetForTraining,feature,number)

n_Featurelist = []            # To store no of features
```

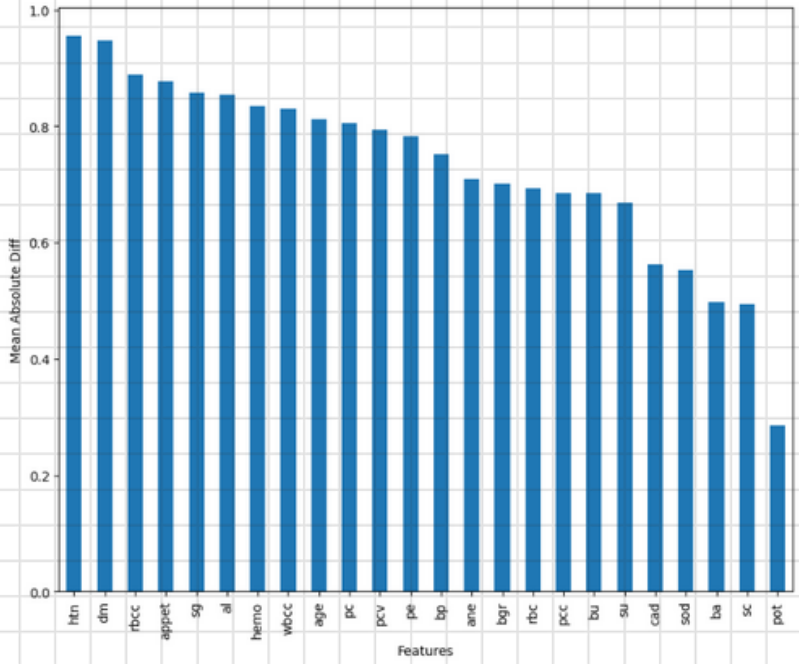
Correlation heat-map of features



Rank graph of features according to different feature selection method

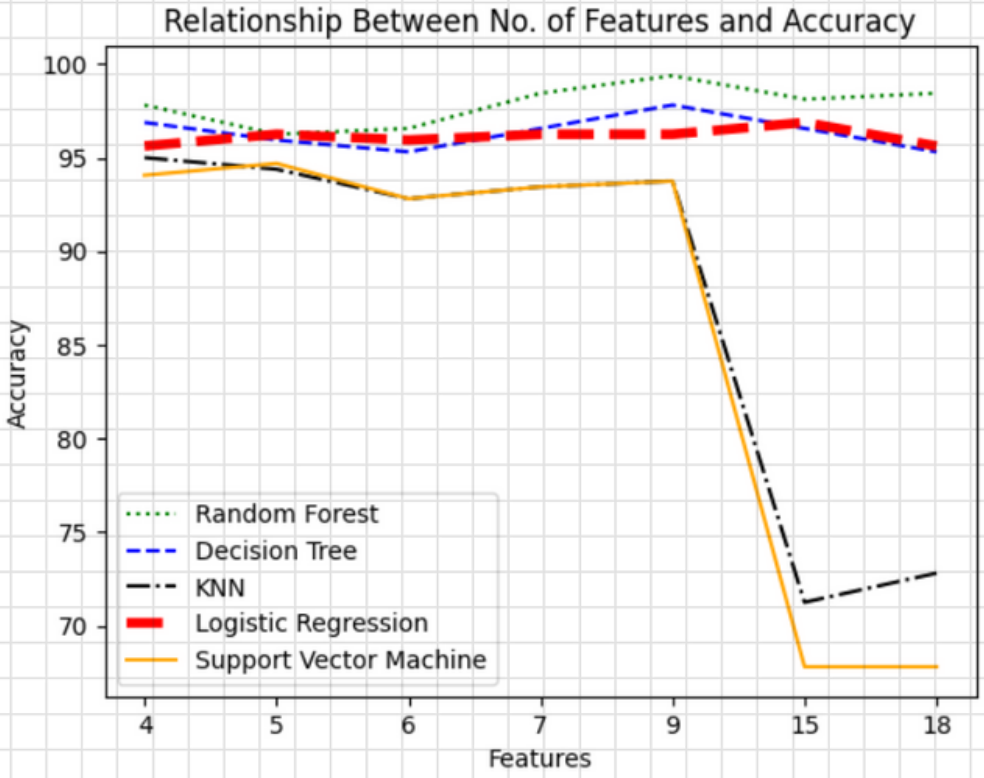


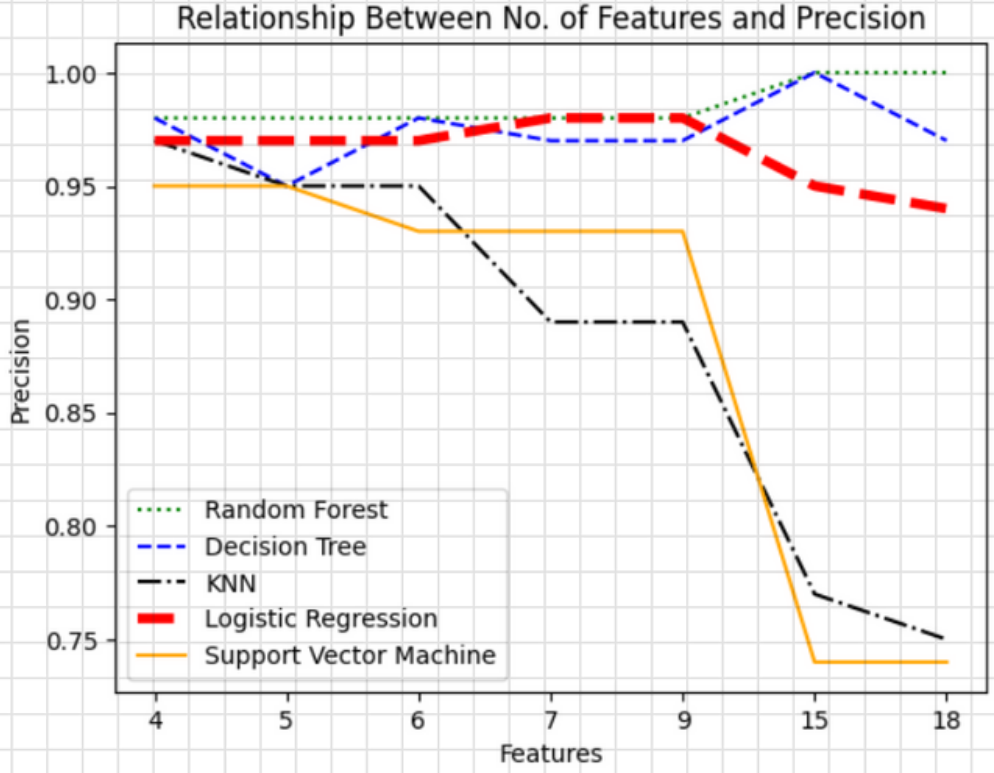
Using Correlation



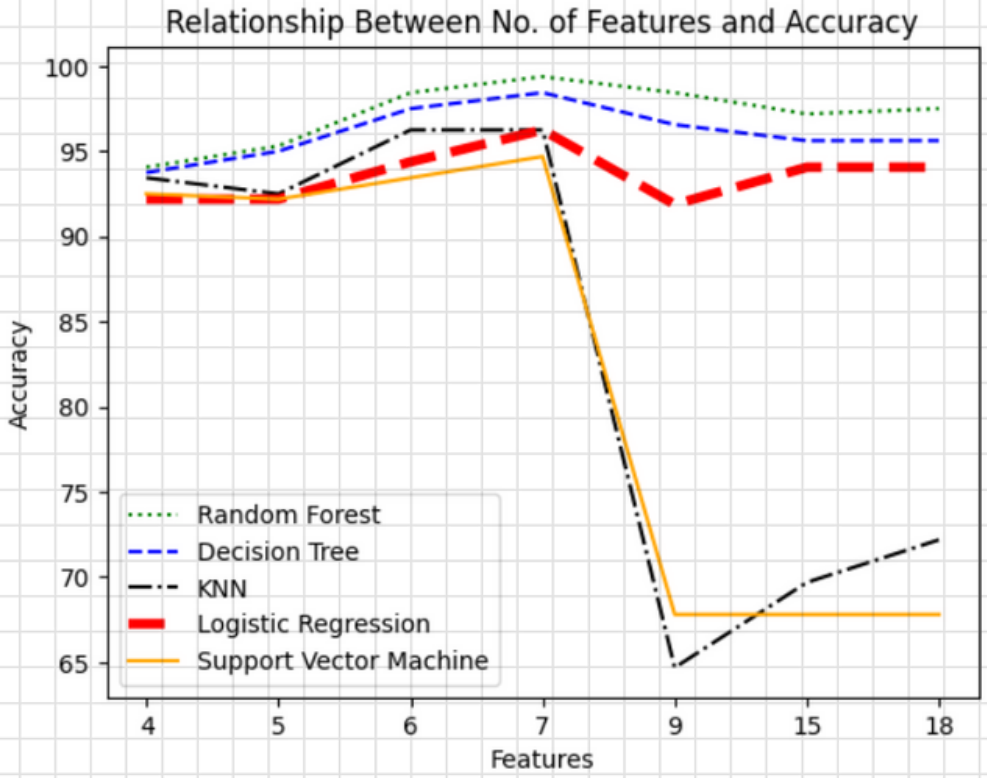
Using Mean Absolute Difference

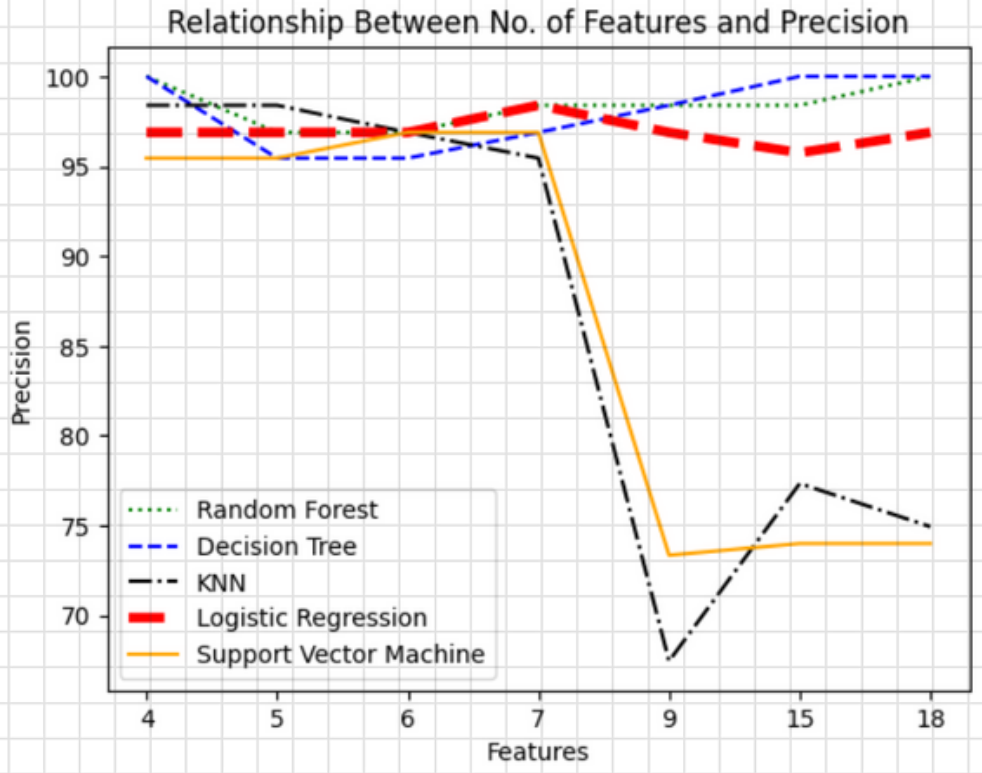
Mode evaluation of different classifiers for correlation method



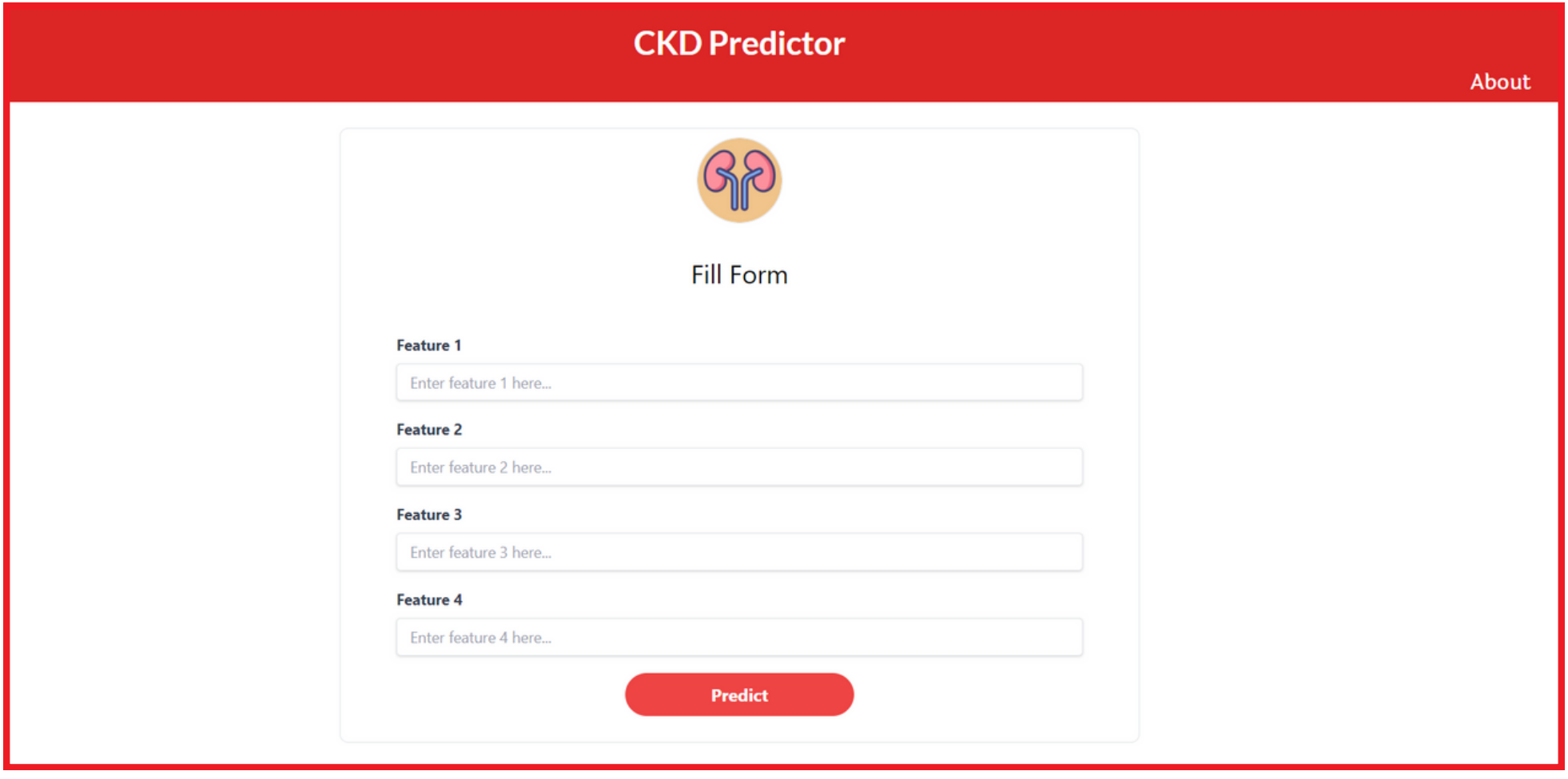


Mode evaluation of different classifiers for Mean absolute difference method





User Interface using React and other web utilities



Work to do

1. Using other feature selection model and comparing results.

We will be using other feature selection algorithms like chi-square, fisher score to see if they provide better result than the used algorithms.
After comparing the different methods on the basis of accuracy, precision and other metrices we will select the best feature selection method.

2. Using ensembling algorithm to combine all models and compare its result.

After coming up with the best feature selection method we will use ensemble method to build the final model. Using ensembling we will be try to improve the performance and accuracy of the models, especially for complex and noisy datasets.

3. Deploying the final model and connecting it with the frontend UI.

We need to deploy the final model to a server and make api so the we can use it to connect to the frontend to make our user interface available for people to use it.

Note: We will also continue doing the literature review side by side for gathering more information

References

1. Centers for Disease Control and Prevention. Chronic Kidney Disease in the United States, 2019. Atlanta, GA: US Department of Health and Human Services, Centers for Disease Control and Prevention; 2019.
2. Elhoseny, M., Shankar, K. & Uthayakumar, J. Intelligent Diagnostic Prediction and Classification System for Chronic Kidney Disease. Sci Rep 9, 9583 (2019).
3. Haratian, A., Maleki, Z., Shayegh, F. et al. Detection of factors affecting kidney function using machine learning methods. Sci Rep 12, 21740 (2022).
4. Bai, Q., Su, C., Tang, W. et al. Machine learning to predict end stage kidney disease in chronic kidney disease. Sci Rep 12, 8377 (2022).
5. Koyner, Jay L. MD; Carey, Kyle A. MPH; Edelson, Dana P. MD, MS; Churpek, Matthew M. MD, MPH, PhD. The Development of a Machine Learning Inpatient Acute Kidney Injury Prediction Model. Critical Care Medicine 46(7):p 1070-1077, July 2018
6. Martha Catalina Morales-Alvarez, Gabriela Garcia-Dolagaray, Andreina Millan-Ferro, Sylvia E. Rosas. Renal Function Decline in Latinos With Type 2 Diabetes, Kidney International Reports, Volume 4, Issue 9 (2019).
7. Prediction and detection models for acute kidney injury in hospitalized older adultsKate R.J., Perez R.M., Mazumdar D., Pasupathy K.S., Nilakantan V. (2016) BMC Medical Informatics and Decision Making, 16 (1) , art. no. 39
8. Machine Learning to Identify Dialysis Patients at High Death RiskAkbilgic O., Obi Y., Potukuchi P.K., Karabayir I., Nguyen D.V., Soohoo M., Streja E., et.al., Kovesdy C.P. (2019) Kidney International Reports, 4 (9) , pp. 1219-1229.
9. Kelly M., Longjohn R., Nottingham K., The UCI Machine Learning Repository, https://archive.ics.uci.edu

Thank You