Chronic Kidney Disease Analysis Using Machine Learning

Calvin Jerome Suares
19BEC0433
Vellore Institute of Technology, Vellore

INDEX

INTRODUCTION TO PROJECT

Chronic kidney disease (CKD) is a significant public health problem worldwide, especially for low and medium income countries. Chronic kidney disease (CKD) means that the kidney does not work as expected and cannot correctly filter blood. About 10% of the population worldwide suffers from (CKD), and millions die each year because they cannot get affordable treatment, with the number increasing in the elderly. According to the Global Burden Disease 2010 study conducted by the International Society of Nephrology, chronic kidney disease (CKD) has been raised as an important cause of mortality worldwide with the number of deaths increasing by 82.3% in the last two decades. Also, the number of patients reaching end-stage renal disease (ESRD) is increasing, which requires kidney transplantation or dialysis to save patients' lives. CKD, in its early stages, has no symptoms; testing may be the only way to find out if the patient has kidney disease. Early detection of CKD in its initial stages can help the patient get effective treatment and then prohibit the progression to ESRD. It is argued that every year, a person that has one of the CKD risk factors, such as a family history of kidney failure, hypertension, or diabetes, get checked. The sooner they know about having this disease, the sooner they can get treatment. To raise awareness and to encourage those who are most susceptible to the disease to perform the tests periodically, we hope that the disease can be detected with the least possible tests and at low cost. So, the objective of this research is to provide an effective model to predict the CKD by least number of predictors.

LITERATURE SURVEY

There is a mounting epidemic of CKD and end-stage renal disease (ESRD) in the US. As of 2002, between 4 million and 20 million Americans were affected with CKD, and about 300 000 were defined as having ESRD or requiring renal replacement therapy. It is estimated that by 2015 the number of patients with ESRD will be 712 000. The total number of expected patients receiving dialysis by 2010 will reach 560 000 resulting in an annual Medicare spending of $28.3 billion by 2010. As of 2007, the total Medicare cost for CKD reached $57.5 billion.

Patients with CKD are at risk for not only progression to ESRD but also increased cardiovascular morbidity and mortality. The key to preventing either of these two outcomes is recognition of the earliest stages of kidney disease and initiation of a targeted and aggressive management plan. The National Kidney Foundation provides evidence-based clinical practice guidelines for all stages of CKD and related complications, which include a recommendation for referral to a nephrologist if CKD is sufficiently advanced. The importance of a timely referral to a nephrologist is evident in multiple studies that have shown an association with late nephrology referral and poor outcomes when starting haemodialysis. Patients with un-recognized CKD may be referred by their provider at a later stage than a patient with recognized CKD.

Only if providers recognize that their patients have CKD will the appropriate targeted management be initiated. Several investigators have demonstrated considerable under-recognition by primary care practitioners. De Lusignan and colleagues demonstrated that less than 4% of patients with CKD had been coded as having renal disease. Studies conducted by manual chart review (bypassing the known *International Classification of Diseases* (ICD)-9 coding sensitivity issues) demonstrated that over three-quarters of patients with CKD were not recognized as having CKD.

A first step in creating a tool to prompt early recognition of CKD is to determine if the provider has recognized the patient's CKD. The tool could search for appropriate documentation of CKD in the patient's notes as a proxy for recognition.

THEORITICAL ANALYSIS

Chronic Kidney Disease (CKD) is a major medical problem and can be cured if treated in the early stages. Usually, people are not aware that medical tests, we take for different purposes could contain valuable information concerning kidney diseases. Consequently, attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease. The information says that it helps us to measure the severity of the problem and we make use of such information to build a machine learning model that predicts Chronic Kidney Disease.

Following are the steps which are to be completed to complete this project:

1. Download the dataset
2. Preprocess or clean the data
    a. Import the libraries
    b. Read the dataset
    c. Analyse the dataset
    d. Drop unnecessary columns
    e. Change the column names
    f. Remove the randomness in the columns
    g. Find the missing values
    h. Handle the missing values
    i. Split the data into independent and dependent variables
    j. Split the data to train and test
3. Train the machine with preprocessed data with an Appropriate Machine learning algorithm to build a model
4. save the model and its dependencies
5. Build a Web application using flask that integrates with Model built.

Model is based on logistic regression, and it obtains the weight of each predictor and a bias. If the sum of the effects of all predictors exceeds a threshold, the category of the sample will be classified as ckd or notckd.

EXPERIMENTAL INVESTIGATION

There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms that you can choose according to the objective that you might have it maybe Classification algorithms or Regression algorithms.

Example: 1. Linear Regression.
2. Logistic Regression.
3. Random Forest Regression / Classification.
4. Decision Tree Regression / Classification.

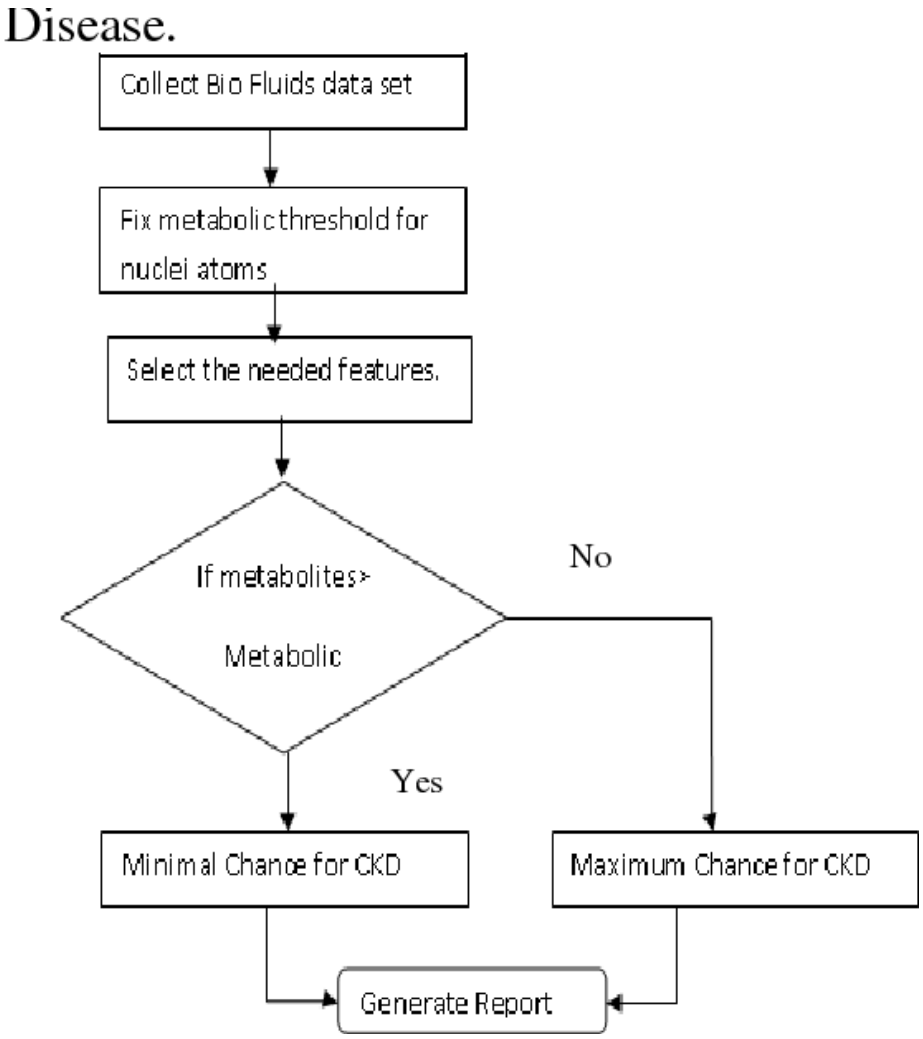
As the prediction for model is classification type, we apply a logistic regression algorithm on our dataset.

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Once the model is trained, it's ready to make predictions.

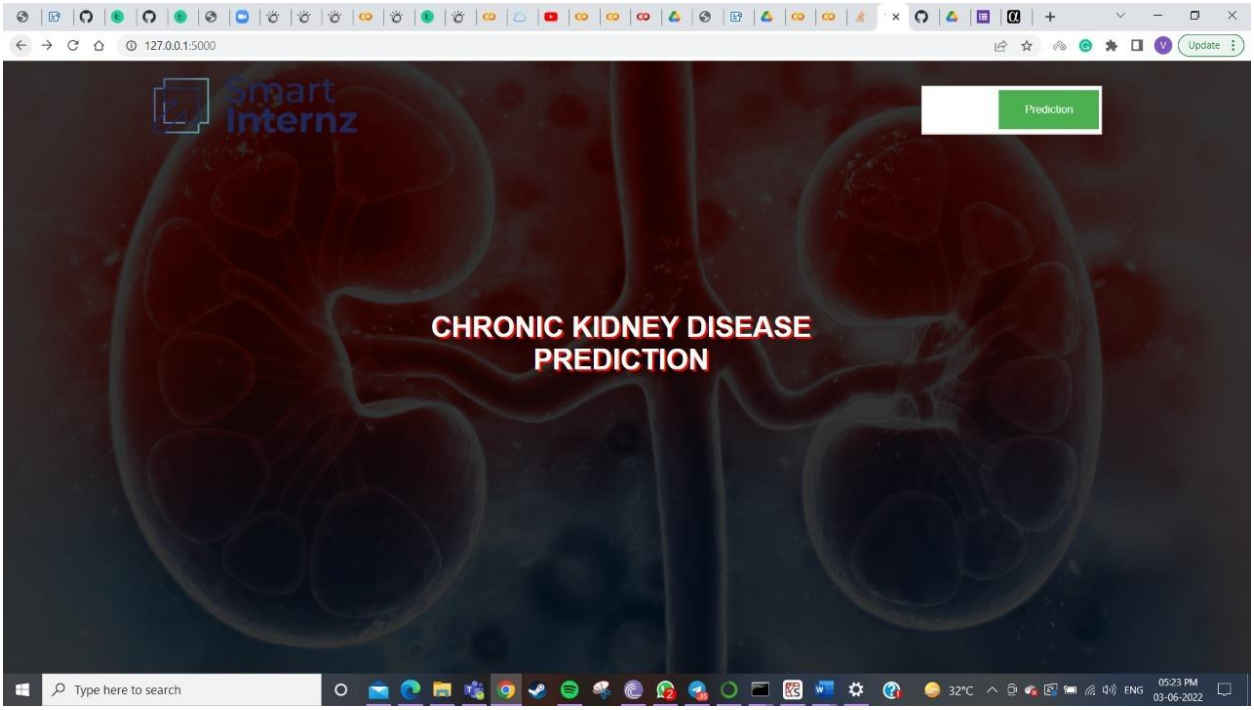
Finally, there is a need to check to see how well our model is performing on the test data. There are many evaluation techniques. For this, we evaluate the accuracy score produced by the model. Confusion Matrix for the model is been used.

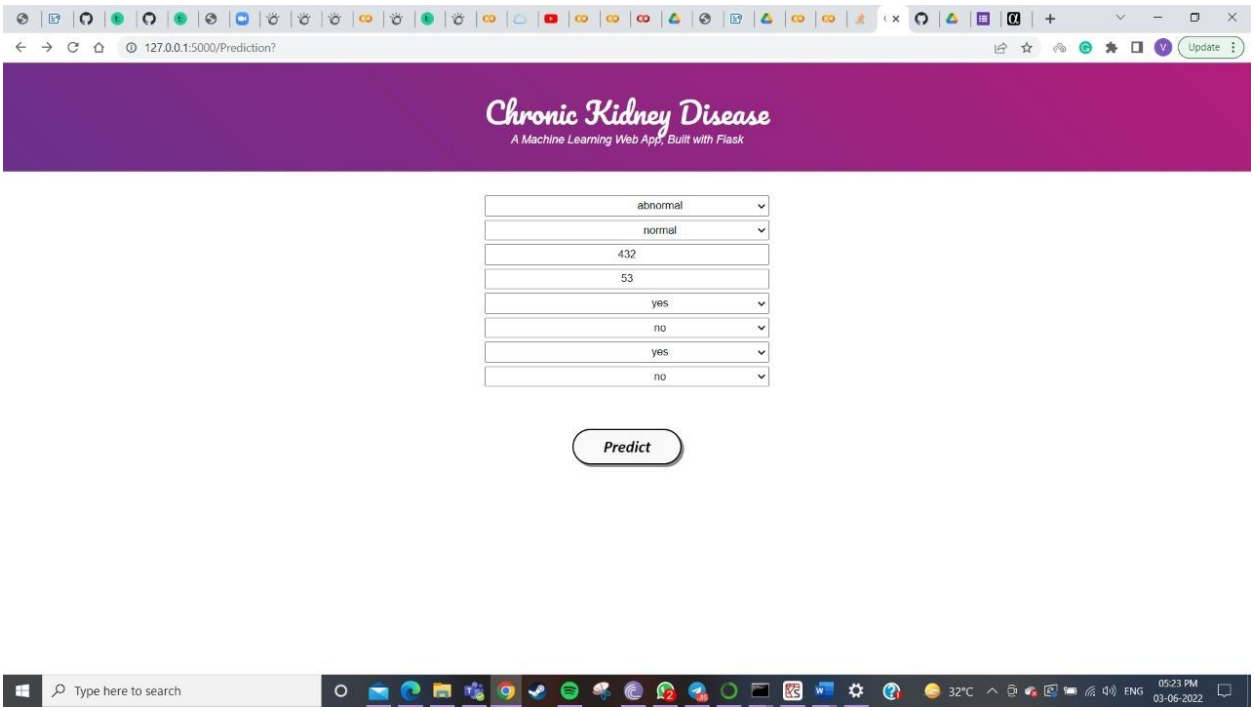
FLOW CHART

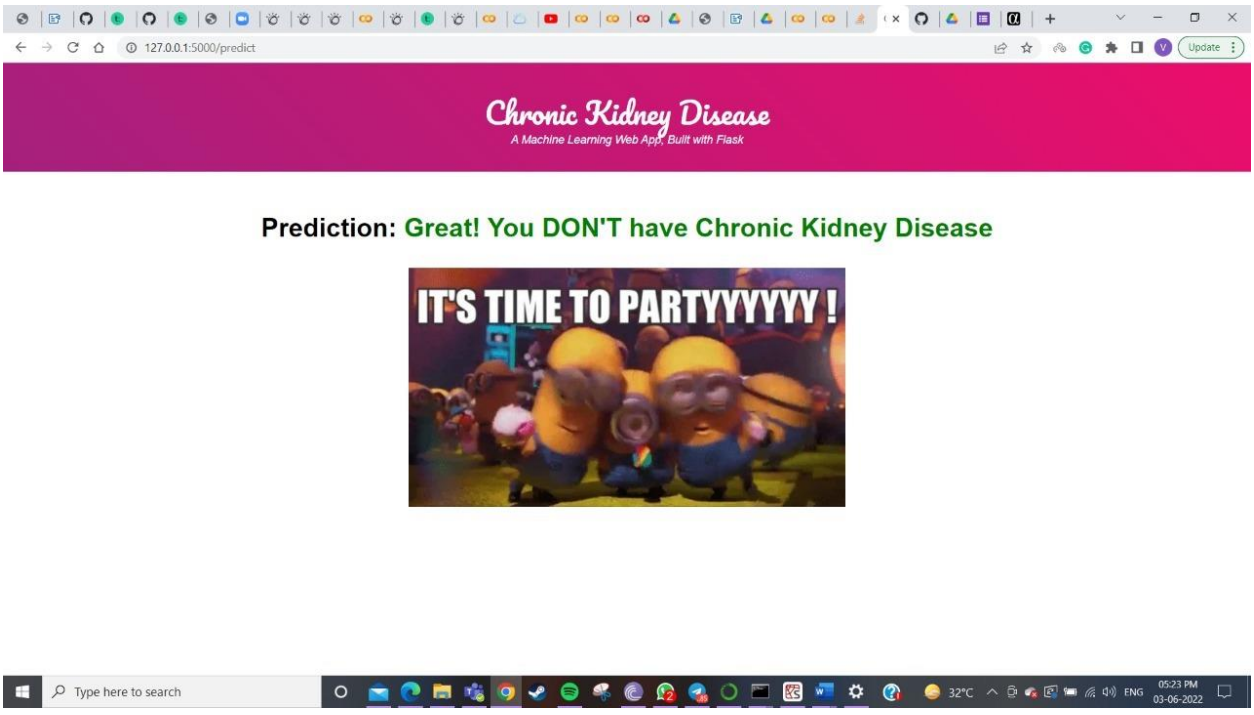

Disease.
Collect Bio Fluids data set
Fix metabolic threshold for nuclei atoms
Select the needed features.
If metabolites> Metabolic
No
Yes
Minimal Chance for CKD
Maximum Chance for CKD
Generate Report

RESULT

The sensitivities of the classifier and word-count methods were 95.4% and 99.8%, respectively. The specificity of both was 99.8%. Categorization of
individual patients as appropriately documented was96.9% accurate. Of 107 patients with manually verified moderate CKD, 32 (22%) lacked appropriate
documentation. Patients whose CKD had not been appropriately documented were significantly less likely to be on renin-angiotensin system inhibitors or have urine protein quantified, and had the illness for half as long (15.1 vs 30.7 months; p<0.01) compared to patients with documentation. The result of each classifier has been evaluated using Confusion Matrix and saved in the form of pickle form in order to use it in web application which is been made with the help of FLASK.




127.0.0.1:5000/Prediction?
Update
Chronic Kidney Disease
A Machine Learning Web App, Built with Flask
abnormal
normal
432
53
yes
no
yes
no
Predict
Type here to search
32°C
ENG
05:23 PM
03-06-2022


127.0.0.1:5000/predict
Update
Chronic Kidney Disease
A Machine Learning Web App, Built with Flask
Prediction: Great! You DON'T have Chronic Kidney Disease
IT'S TIME TO PARTYYYYYY !
Type here to search
32°C
ENG
05:23 PM
03-06-2022

ADVANTAGES AND DISADVANTAGES

Advantages:

Logistic regression analysis is a statistical technique that describes the relationship between an independent variable (either continuous or not) and a dichotomy dependent variable (or dummy variable) (that is, a variable with only two possible values: 0 = outcome absent and 1 = outcome present). Hence, it eases the process.

The only limitation for this project is the dataset which is considered is quite smaller in size.

APPLICATIONS

1. Chronic Kidney Disease (CKD) is a major medical problem which has similar ramification as cancer.
2. It can be cured if treated in the early stages.
3. Attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease.
4. This machine learning model is applicable in medical science.

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 diagnoses of diseases; hence, it might be a promising method for diagnosing CKD. It has become a new medical tool with the development of information technology and has a broad application prospect because of the rapid development of electronic health records.

CONCLUSION AND FUTURE SCOPE

This work examines the ability to detect CKD using machine learning algorithms while considering the least number of tests or features. To approach this aim we have used logistic regression which is a machine learning classifier. In order to reduce the number of features and remove redundancy, the association between variables has been studied. Through this guided project I learnt how to implement Logistic regression in the real world problem and also understood the problem to classify if it is a regression or a classification problem. Apart from this, it also provided in-depth knowledge about how to handle missing values and enlightened me with the information regarding Label encoding. I also learnt about Pickle library with which I was not familiar before.

After completing this, we can add more categories in this work, can make this more efficient. Using more classifiers on this dataset can get a better understanding on which classifier can be the best for this work.

Video Link:

https://drive.google.com/file/d/1KrMLGiI3JNlX_K_zDLsD6VdZazTMAjoB/view?usp=sharing

Appendix:

```
import pandas as pd
import numpy as np
df=pd.read_csv('chronickidneydisease.csv')
pd.set_option('display.max_columns', None)

df.head()
df.shape
df.info()
df.isnull().sum()
df.drop(columns=['id'],axis=1,inplace=True)
df.columns=['age','blood_pressure','specific_gravity','albumin',
              'sugar','red_blood_cells','pus_cell','pus_cell_clumps','bact
eria',
              'blood glucose random','blood_urea','serum_creatinine','sodi
um','potassium',
              'hemoglobin','packed_cell_volume','white_blood_cell_count','
red_blood_cell_count',
              'hypertension','diabetesmellitus','coronary_artery_disease',
'appetite',
              'pedal_edema','anemia','class'] # manually giving the name
of the columns
df.columns
cat_df=df.select_dtypes('object')
cat_df.head()
# Replacing null values in cat_df by mode
cat_df = cat_df.fillna(cat_df.mode().iloc[0])
cat_df.head()
cat_df.shape
cat_df.isnull().sum()
cat_df.head()
for column in cat_df.columns:                                    # value_count
s Loop
    print("\n" + column)
    print(cat_df[column].value_counts())
cat_df['class']=cat_df['class'].replace("ckd\t","ckd") #replace is used fo
r renaming
cat_df['class'].value_counts()
cat_df['coronary_artery_disease']=cat_df['coronary_artery_disease'].replac
e("\tno","no") #replace is used for renaming
cat_df['coronary_artery_disease'].value_counts()
cat_df['diabetesmellitus']=cat_df['diabetesmellitus'].replace("\tno","no")
cat_df['diabetesmellitus']=cat_df['diabetesmellitus'].replace("\tyes","yes
")
```

```
cat_df['diabetesmellitus']=cat_df['diabetesmellitus'].replace(" yes","yes"
)
cat_df['diabetesmellitus'].value_counts()
cat_df['red_blood_cell_count']=cat_df['red_blood_cell_count'].replace("\t?
","5.2")
cat_df['red_blood_cell_count'].value_counts()
cat_df['white_blood_cell_count']=cat_df['white_blood_cell_count'].replace(
"\t?","9800")
cat_df['white_blood_cell_count'].value_counts()
cat_df['packed_cell_volume']=cat_df['packed_cell_volume'].replace("\t?","4
1")
cat_df['packed_cell_volume']=cat_df['packed_cell_volume'].replace("\t43","
41")
cat_df['packed_cell_volume'].value_counts()
cat_df.head()
cat_df.info()
cat_df.isnull().sum()
      from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
cat_df=cat_df.apply(le.fit_transform)
cat_df.head()
cat_df.dtypes
num_df=df.select_dtypes('float64','int64')
num_df.head()
num_df.shape
num_df.isnull().sum()
num_df = num_df.fillna(num_df.median())
num_df.head()
num_df.isnull().sum()
num_df.info()
cat_df_col_null=cat_df.columns[cat_df.isnull().any()==True].tolist()
cat_df[cat_df_col_null].isnull().sum()
cat_df[cat_df_col_null].isnull().sum().count()
new_df=num_df.join(cat_df)
new_df.head()
new_df.shape
new_df.isnull().sum()
new_df.info()
new_df.corr()
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(30,27))

sns.heatmap(new_df.corr(),annot=True)
new_df.corr()['class'].sort_values(ascending=False)
```

```
X1=new_df.drop(['class'],axis=1)
X1.head()
X1.shape
from sklearn.preprocessing import scale
X_scaled=pd.DataFrame (scale(X1), columns=X1.columns)
X_scaled.head()
from sklearn.decomposition import PCA
pca=PCA()
X_pca=pd.DataFrame(pca.fit_transform(X_scaled))
pca.explained_variance_ratio_
import matplotlib.pyplot as plt
import numpy as np
plt.bar(np.arange(24),pca.explained_variance_ratio_)
pca.explained_variance_ratio_[:10].sum()
pca.explained_variance_ratio_.sum()
X=new_df.loc[:,['age','bacteria','white_blood_cell_count','potassium','red
_blood_cells',
                'sodium','pus_cell','red_blood_cell_count','packed_cell_vol
ume','specific_gravity','hemoglobin']]
X.head()
from sklearn.preprocessing import scale
scaled_X=pd.DataFrame (scale(X), columns=X.columns)
scaled_X.head()
scaled_X.shape
new_df.loc[new_df['class']==0, 'class'] = 'disease'
new_df.loc[new_df['class']==1, 'class'] = 'no_disease'
y=new_df['class']
y.head()
y.value_counts()
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(scaled_X,y,test_size=0.2,ra
ndom_state=42)
X_train.shape
X_test.shape
from sklearn.ensemble import RandomForestClassifier
model1=RandomForestClassifier()
model1.fit(X_train,y_train)
y_predict1=model1.predict(X_test)
from sklearn.metrics import accuracy_score,classification_report
rfc=accuracy_score(y_test,y_predict1)
rfc
pd.crosstab(y_test,y_predict1)
print(classification_report(y_test,y_predict1))
print('Train accuracy score of RFC:',model1.score(X_train, y_train))
print('Test accuracy score of RFC:',model1.score(X_test, y_test))
```

```
from sklearn.linear_model import LogisticRegression
model2=LogisticRegression()
model2.fit(X_train,y_train)
y_predict2=model2.predict(X_test)
lg=accuracy_score(y_test,y_predict2)
lg
pd.crosstab(y_test,y_predict2)
print(classification_report(y_test,y_predict2))
print('Train accuracy score of LOG_RE:',model2.score(X_train, y_train))
print('Test accuracy score of LOG_RE:',model2.score(X_test, y_test))
probability = model1.predict_proba(X_test)[:, 1]
from sklearn.metrics import roc_curve,roc_auc_score
import matplotlib.pyplot as plt
fpr,tpr,thresholds = roc_curve(y_test,probability)


plt.plot(fpr,tpr)
plt.title('ROC Curve')
plt.xlabel('FPR')
plt.ylabel('TPR')
plt.show()
roc_auc_score(y_test,probability)
import pickle
pickle.dump(model1, open('ckd1.pkl','wb'))
pwd
```