

Team ID PNT2022TMID45417

Project Name :Early Prediction of Chronic Kidney Disease

SOURCE CODE

```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
os.getcwd()
path='C:\\Users\\pc\\downloads\\'
data=pd.read_csv(path+'chronickidneydisease.csv')
data.head(10)
data.tail(10)
data.shape
data.columns=['id','age','blood_pressure','specific_gravity','albumin','sugar','red_blood_cells',
              'pus_cell','pus_cell_clumps','bacteria','blood glucose random',
              'blood_urea','serum_creatinine','sodium','potassium','hemoglobin','packed_cell_volume',
              'white_blood_cell_count', 'red_blood_cell_count',
              'hypertension','diabetesmellitus','coronary_artery_disease',
              'apettite','pedal_edema','anemia','class']
data.columns
data.info()
data.drop(['id'],axis=1,inplace=True)
data
#target column
data['class'].unique()
#rectify target column
data['class']=data['class'].replace('ckd\\t','ckd')
data['class'].unique()
#fetching categorical column
cat=data.select_dtypes(include=['object']).columns.tolist()
cat
#removing column which are not categorical
cat.remove('red_blood_cell_count')
cat.remove('packed_cell_volume')
cat.remove('white_blood_cell_count')
cat
```

```

num=data.select_dtypes(include=['float64']).columns.tolist()#fetch numerical column
num.remove('specific_gravity')#remove which are not numerical
num.remove('albumin')
num.remove('sugar')
num
#adding column which is numerical
num.append('red_blood_cell_count')
num.append('packed_cell_volume')
num.append('white_blood_cell_count')
num
sns.pairplot(data,hue='class')
fig=plt.figure(figsize=(10,5))
fig
sns.barplot(x='blood glucose random',y='class',data=data)
cat.append('specific_gravity')#adding column which is categorical
cat.append('albumin')
cat.append('sugar')
cat
a=data['coronary_artery_disease'].unique()
#b=data['sugar'].unique()
#c=data['albumin'].unique()
#d=data['specific_gravity'].unique()
#e=data['anemia'].unique()
#f=data['pedal_edema'].unique()
#g=data['apettite'].unique()
h=data['diabetesmellitus'].unique()
#i=data['bacteria'].unique()
#j=data['hypertension'].unique()
#k=data['red_blood_cell_count'].unique()
#l=data['pus_cell'].unique()
#m=data['pus_cell_clumps'].unique()
a,h
#rectifying the categorical column classes
data['coronary_artery_disease']=data['coronary_artery_disease'].replace('\tno','no')
data['coronary_artery_disease'].unique()
data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'yes':'yes','tyes':'yes','\tno':'no'})
data['diabetesmellitus'].unique()

#handling missing value

data.isna().sum()

#before handling the numeric variable which is considered as string should be convert to numerical

data.red_blood_cell_count=pd.to_numeric(data.red_blood_cell_count,errors='coerce')

```

```

data.packed_cell_volume=pd.to_numeric(data.packed_cell_volume,errors='coerce')
data.white_blood_cell_count=pd.to_numeric(data.white_blood_cell_count,errors='coerce')
#handle numerical column null values
data['blood_pressure'].fillna(data['blood_pressure'].mean(),inplace=True)
data['blood_urea'].fillna(data['blood_urea'].mean(),inplace=True)
data['blood glucose random'].fillna(data['blood glucose random'].mean(),inplace=True)
data['serum_creatinine'].fillna(data['serum_creatinine'].mean(),inplace=True)
data['sodium'].fillna(data['sodium'].mean(),inplace=True)
data['potassium'].fillna(data['potassium'].mean(),inplace=True)
data['hemoglobin'].fillna(data['hemoglobin'].mean(),inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['age'].fillna(data['age'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['red_blood_cell_count'].fillna(data['red_blood_cell_count'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['white_blood_cell_count'].fillna(data['white_blood_cell_count'].mode()[0],inplace=True)
data['packed_cell_volume'].fillna(data['packed_cell_volume'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=True)
data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)
data['coronary_artery_disease'].fillna(data['coronary_artery_disease'].mode()[0],inplace=True)
data['apettite'].fillna(data['apettite'].mode()[0],inplace=True)
data['pedal_edema'].fillna(data['pedal_edema'].mode()[0],inplace=True)
data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)
data['specific_gravity'].fillna(data['specific_gravity'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data.isnull().sum()
from sklearn.preprocessing import LabelEncoder
for i in cat:

```

```

print('label of encoder= ',i)
lei=LabelEncoder()
print(data[i])
data[i]=lei.fit_transform(data[i])
print(data[i])
print('***100)

data.corr().T
selcols=['red_blood_cells','pus_cell',
'diabetesmellitus', 'coronary_artery_disease','blood_urea','pedal_edema','anemia',
'blood glucose random']
x=pd.DataFrame(data,columns=selcols)
y=pd.DataFrame(data,columns=['class'])
print(x.shape)
print(y.shape)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
from sklearn.linear_model import LogisticRegression
lgr=LogisticRegression()
lgr.fit(x_train,y_train)
y_pred=lgr.predict(x_test)
y_pred1=lgr.predict([[140,45,0,0,0,0,0,0]])
print(y_pred1)
from sklearn.metrics import accuracy_score
acc=accuracy_score(y_test,y_pred)
acc
from sklearn.metrics import confusion_matrix
conf_mat=confusion_matrix(y_test,y_pred)
conf_mat
import pickle
pickle.dump(lgr,open('CKD.pkl','wb'))

```

