

PYTHON CODE

| | |
|---------------------|--|
| Team Id | PNT2022TMID17480 |
| Project Name | Early Detection of Chronic Kidney Disease Using Machine Learning |

```
# -*- coding: utf-8 -*-  
"""CKD Model.ipynb
```

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1SImgoFXHBhtBOe8G5_afIdpxAgUlgnD3

```
**Import Packages**  
"""
```

```
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import tensorflow  
from tensorflow import keras as tf  
from keras.layers import Dense, Activation  
from keras import Sequential  
from keras.models import load_model as tl  
from keras.optimizers import Adam
```

```
"""**Read dataset**"""
```

```
data = pd.read_csv("/content/drive/MyDrive/Dataset_CKD.csv")  
print(data)
```

```
"""**Understanding Data Type and Features**"""
```

```
print(data.info())
```

```
"""**Handling Missing Values**
```

```
Remove null values
```

```
"""
```

```
data=data.dropna(how="any")  
print(data)
```

```
"""**Label Encoding** (String values to Numeric values)"""
```

```
data['rbc'] = data['rbc'].map({"abnormal":1,"normal":0})  
data['pc'] = data['pc'].map({"abnormal":1,"normal":0})  
data['pcc'] = data['pcc'].map({"present":1,"notpresent":0})  
data['ba'] = data['ba'].map({"present":1,"notpresent":0})  
data['htn'] = data['htn'].map({"yes":1,"no":0})  
data['dm'] = data['dm'].map({"yes":1,"no":0})  
data['cad'] = data['cad'].map({"yes":1,"no":0})  
data['pe'] = data['pe'].map({"yes":1,"no":0})  
data['ane'] = data['ane'].map({"yes":1,"no":0})  
data['appet'] = data['appet'].map({"poor":1,"good":0})  
data['classification'] = data['classification'].map({"ckd":1,"notckd":0})  
data['pcv'] = data['pcv'].astype('int')  
data['wc'] = data['wc'].astype('int')  
data['rc'] = data['rc'].astype('float')  
print(data)
```

```
"""**Splitting Dependent and Independent Variable**"""
```

```
X = data.iloc[:,1:25].values
y = data.iloc[:, 25].values
```

```
"""**Split Train and Test set**"""
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30,
random_state = 121)#101
print("X train value")
print(X_train)
print("Y train value")
print(y_train)
```

```
"""**Model Building**"""
```

```
model = tf.Sequential()
model.add(tf.layers.Dense(64,input_dim=24,activation='relu'))
```

```
"""**Hidden and Output Layer**"""
```

```
model.add(tf.layers.Dense(128,activation='relu'))
model.add(tf.layers.Dense(256,activation='relu'))
model.add(tf.layers.Dense(512,activation='relu'))
model.add(tf.layers.Dense(1,activation='sigmoid'))
```

```
"""**Compile Model**"""
```

```
model.compile(loss="binary_crossentropy",optimizer='adam',metrics=['
accuracy'])
model.fit(X_train,y_train,epochs=1000)
```

```
""" **Save Model**"""
```

```
model.save("/content/drive/MyDrive/train.h5")
```

```
"""**Prediction**"""
```

```
ypred=model.predict(X_test)  
ypred = ypred.round()  
print(ypred)  
print(y_test)
```

```
"""**Model Evaluation**"""
```

```
from sklearn.metrics import confusion_matrix  
cm = confusion_matrix(y_test,ypred)  
print(cm)
```