```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

df=pd.read_csv("/content/BostonHousing.csv").values

x=df[:,0:-1]
y=df[:,-1]

test_split=float(input("Enter the test_size in 0 to 1 :- "))

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=test_split)

dist=np.zeros(shape=x_test.shape[0])
pred=np.zeros(shape=x_test.shape[0])

k=int(input("Enter the number of nearest neighbours to be used K :- "))

for i in range(x_test.shape[0]):
  dist=np.sqrt(np.sum((x_train-x_test[i])**2,axis=1))
  kminind=np.argpartition(dist,k)[0:k]
  invdist=1/(dist+10e-20)
  denom=sum(invdist[kminind])
  pred[i]=np.dot(invdist[kminind]/denom,y_train[kminind])

#print(pred)

#mean absolute error
def MAE(pred,y_test):
  return np.mean(abs(pred-y_test))

#mean squared error
def MSE(pred,y_test):
  return np.mean((pred-y_test)**2)

#mean absolute percentage error

def MAPE(pred,y_test):
  return (np.mean(abs((pred-y_test)/y_test)))*100

def RMSE(pred,y_test):
  a=np.mean((pred-y_test)**2)
  return np.sqrt(a)


print("MAE",MAE(pred,y_test))
print("MSE",MSE(pred,y_test))
print("MAPE",MAPE(pred,y_test))
print("RMSE",RMSE(pred,y_test))
```

```
Enter the test_size in 0 to 1 :- 0.2
Enter the number of nearest neighbours to be used K :- 2
MAE 4.000134568050943
MSE 38.14260085297525
```

```
MAPE 17.975317398170567
RMSE 6.1759696285664525
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error , mean_squared_error
from sklearn.neighbors import KNeighborsRegressor

df=pd.read_csv("/content/BostonHousing.csv",header='infer').values

x=df[:,0:-1]
y=df[:,-1]

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

k=2

model=KNeighborsRegressor(n_neighbors=k,weights="distance")
model.fit(x_train,y_train)
pred=model.predict(x_test)

mae=mean_absolute_error(y_test,pred)
mse=mean_squared_error(y_test,pred)
rmse=np.sqrt(mse)
mape=np.mean(abs((y_test-pred)/y_test))*100

print("MAE",mae)
print("MSE",mse)
print("MAPE",mape)
print("RMSE",rmse)
```

```
MAE 4.822186620759524
MSE 49.633702405135885
MAPE 23.367797073949742
RMSE 7.045119048329552
```

✓ 0s completed at 6:33 PM