EX NO:2a DATE:10.07.2023	**Implement a simple problem like regression in keras**

AIM:

To write a program to implement the simple problem like regression in keras.

PROCEDURE:

- Import the dataset into the python environment and load the dataset.
- Check the null values and create a model.
- To split the test and train data using scikit learn package.
- To install the tensorflow package in the python environment.
- Create a model using the tensorflow and import keras package.
- To compile the model and fit with the correct epochs.
- To predict the model and check the accuracy score and plot the line.

PROGRAM:

Importing packages and load the dataset

```
import pandas as pd
data=pd.read_csv('test.csv')
data.head()
```

Output:

	x	y
0	77	79.775152
1	21	23.177279
2	22	25.609262
3	20	17.857388
4	36	41.849864

Splitting the train and testing data

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(data['x'],data['y'],test_size=0.75)
```

Creating a model using keras package

```
from tensorflow import keras
model=keras.Sequential()
model.add(keras.layers.Dense(100,input_dim=1,activation='relu'))
model.add(keras.layers.Dense(200,input_dim=100,activation='relu'))
model.add(keras.layers.Dense(200,input_dim=200,activation='relu'))
model.add(keras.layers.Dense(1,input_dim=200))
```

summary of the model

```
model.summary()
```

Output:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 100)               200

dense_1 (Dense)              (None, 200)               20200

dense_2 (Dense)              (None, 200)               40200

dense_3 (Dense)              (None, 1)                 201

=================================================================
Total params: 60,801
Trainable params: 60,801
Non-trainable params: 0
_________________________________________________________________
```

Compile and fit the model

```
model.compile(optimizer='adam',loss='mean_squared_error',metrics='mse')
model.fit(X_train,y_train,epochs=100)
```

Output:

```
Epoch 1/100
3/3 [==============================] - 1s 8ms/step - loss: 3763.2336 - mse: 3763.2336
Epoch 2/100
3/3 [==============================] - 0s 5ms/step - loss: 2494.8018 - mse: 2494.8018
Epoch 3/100
3/3 [==============================] - 0s 5ms/step - loss: 1547.8860 - mse: 1547.8860
Epoch 4/100
3/3 [==============================] - 0s 5ms/step - loss: 784.8910 - mse: 784.8910
Epoch 5/100
3/3 [==============================] - 0s 5ms/step - loss: 233.6637 - mse: 233.6637
Epoch 6/100
3/3 [==============================] - 0s 5ms/step - loss: 23.8476 - mse: 23.8476
Epoch 7/100
3/3 [==============================] - 0s 5ms/step - loss: 175.3268 - mse: 175.3268
Epoch 8/100
3/3 [==============================] - 0s 5ms/step - loss: 251.2844 - mse: 251.2844
Epoch 9/100
3/3 [==============================] - 0s 6ms/step - loss: 123.8949 - mse: 123.8949
Epoch 10/100
3/3 [==============================] - 0s 5ms/step - loss: 18.0946 - mse: 18.0946
Epoch 11/100
3/3 [==============================] - 0s 5ms/step - loss: 31.5778 - mse: 31.5778
Epoch 12/100
3/3 [==============================] - 0s 5ms/step - loss: 66.2557 - mse: 66.2557
Epoch 13/100
3/3 [==============================] - 0s 5ms/step - loss: 55.4989 - mse: 55.4989
Epoch 14/100
3/3 [==============================] - 0s 5ms/step - loss: 24.6223 - mse: 24.6223
```

Predict the model

```
y_pred=model.predict(X_test)
```

Output:

```
8/8 [==============================] - 0s 2ms/step
```

Check the accuracy for the model

```
from sklearn.metrics import r2_score
r2_score(y_pred,y_test)
```

Output:

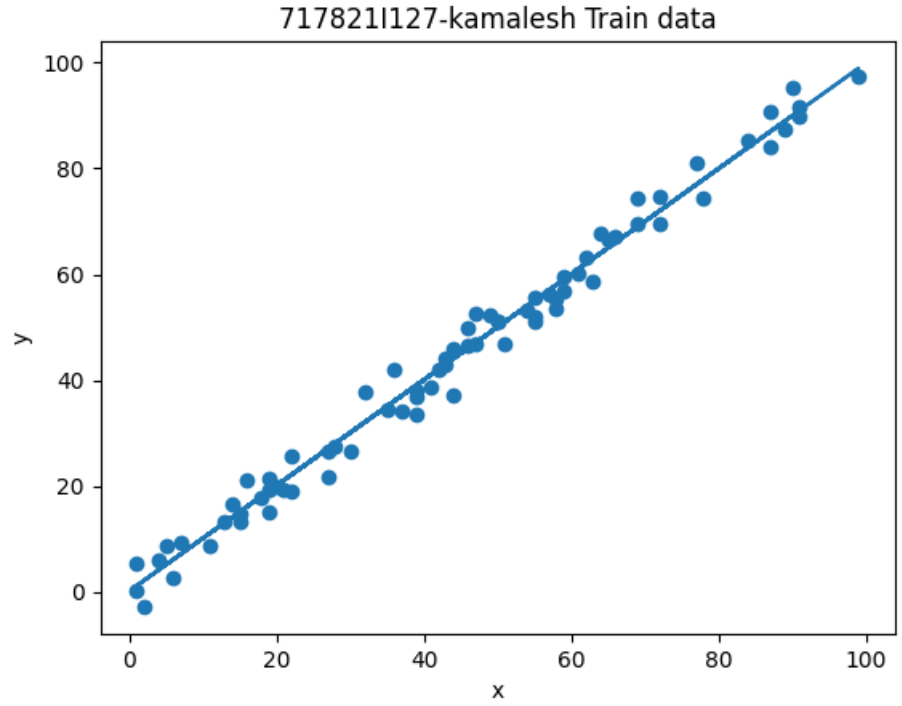
```
0.987754169014235
```

Plot the train data

```
import matplotlib.pyplot as plt
plt.scatter(X_train,y_train)
plt.plot(X_train,model.predict(X_train))
plt.title("717821I127-kamalesh Trained data")
plt.xlabel("x")
plt.ylabel("y")
```

Output:

```
3/3 [==============================] - 0s 3ms/step
Text(0, 0.5, 'y')
```

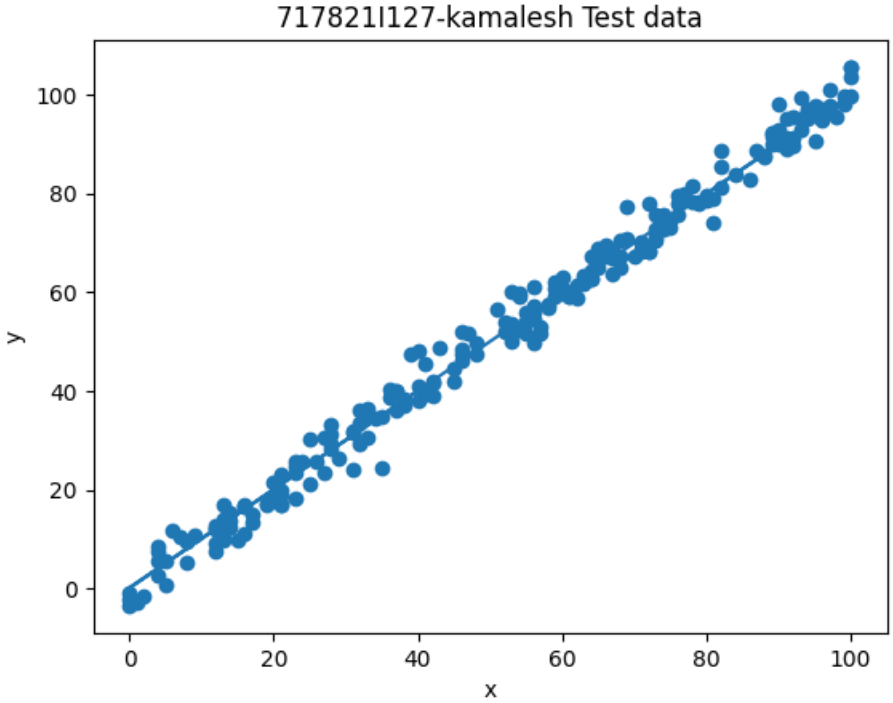


Plot the test data

```
plt.scatter(X_test,y_test)
plt.plot(X_test,model.predict(X_test))
plt.title("717821I127-kamalesh Test data")
plt.xlabel("x")
plt.ylabel("y")
```

Output:

```
8/8 [==============================] - 0s 2ms/step
Text(0, 0.5, 'y')
```



Preparation	30	
Lab Performance	30	
Report	40	
Total	100	
Initial of Faculty		

RESULT:

Thus, the implementation of simple program like regression using keras is successfully executed.