

Price Prediction

Programmed by Sai Yashaswi -

```
# Importing necessary modules from TensorFlow and NumPy
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from tensorflow.keras.losses import MeanSquaredError
from tensorflow.keras.optimizers import Adam
import numpy as np
import tensorflow as tf

# Asking user for inputs related to house construction
price = int(input("Enter the price of the land in your area:\n"))
bed_rooms = int(input("Enter the no.of bedrooms you want:\n"))
floors = int(input("Enter the no.of floors you want:\n"))
other_charges = int(input("Enter the other charges like paint etc:\n"))

# Summing up all inputs to form the target output (label)
total_cost = price + bed_rooms + floors + other_charges

# Creating input feature array
x_in = np.array([[price, bed_rooms, floors, other_charges]])

# Defining a simple Sequential model with 2 hidden layers
model = Sequential([
    Dense(units=25, activation='relu'), # First hidden layer with ReLU activation
    Dense(units=15, activation='relu'), # Second hidden layer with ReLU activation
    Dense(units=1, activation='linear') # Output layer (regression)
])

# Compiling the model with MSE loss and Adam optimizer
model.compile(
    loss=MeanSquaredError(),
    optimizer=Adam()
)

# Creating the label array and specifying data type as float32 (important for TF models)
y = np.array([total_cost], dtype=np.float32)

# Checking model output before training (initial prediction)
print(model(x_in))

# Training the model on the single input sample
model.fit(x_in, y, epochs=200)

# Predicting final price after training and printing it
print(f"Final selling price of your house is {round(model.predict(x_in)[0][0])}")
```