

Group Assignment

Rajesh Kazankodath EPGP-11-91

David Boodala EPGP-11-031

Robin Varghese EPGP-11-099

Shantanu Kamnapure EPGP-11-

Sagar Salunkhe EPGP-11-

Shweta Sude EPGP-11-

Simranjeet Singh Kahlon EPGP-11-

Subose Pai EPGP-11-117

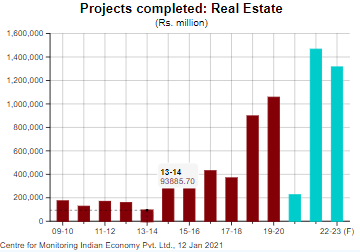
Mentored by Prof. A. K. Swain

Artificial Intelligence for Business

Assignment – Real Estate Price Predictor

**PROJECT INTRODUCTION:**

The real estate project completions especially the “housing construction” is expected to spike in 2020-21 as projected by CMIE (Centre for Monitoring Indian Economy Pvt Ltd) in January 2021. An estimated Rs 1.5 trillion worth of projects are expected to be completed in the year 2021-22 in comparison to only Rs 226 million worth of project scheduled for completion in 2020-21.



This clearly indicates that the real estate sector would continue to attract investors with various projects. With government having taken measures like - infusing liquidity, reducing of levies on realty developers in the form of premiums for a limited period and measures like stamp duty cuts could eventually improve the cash flow and help aid completion of several projects. What historically is a matter of concern to developers and home buyers is arriving at the right price for the property based on several distinct features of the property so that the deal is adjusted to the fair price. Predicting the price however is not easy unless predictor models are appropriately designed.

This assignment takes into consideration several key features of the property to predict the fair price of properties spread over various locations in Pune region. The model is built using Artificial Neural Network and relies on the historical data available in real estate website [www.magicbricks.com](http://www.magicbricks.com). The data for training has been downloaded using the python programme and therefore a large amount of data was made available for the training of ANN as part of this assignment. Some of the key features that were considered as the input variables are location, number of bedrooms, balconies, parking, bathrooms and carpet area.

PHASES OF TRAINING THE MODEL

|  |  |  |
| --- | --- | --- |
| Phase A: | Setting up ANN architecture | Define ANN Architecture |
|  |  | Initialize ANN Weights |
| Phase B: | Forward Propagation | Pass inputs through the ANN |
|  |  | Compare the predicted output with the actual price of the property in the training data set |
| Phase C: | Backpropagation (training) | Compute the cost (Cost of Error) |
|  |  | Update the weights in the ANN |
|  |  | Define a stopping criterion |

**Data Extraction**

We have collected data from [www.magicbricks.com](http://www.magicbricks.com/) for Pune area.

We have downloaded the data from the following Website location

<https://www.magicbricks.com/property-for-sale/residential-real-estate?proptype=Multistorey-Apartment,Builder-Floor-Apartment,Penthouse,Studio-Apartment,Residential-House,Villa&cityName=Pune>

The code for downloading has been written in Python language. The code is available at **webscraping.py**

The downloaded data is in the comma separated format and has the following Features.

Below variable are Numerical Variables

Location : The locality in pune that this property is located

Bedroom : Number of Bedroom available

Balcony : Number of Balconies available

park\_count : Number of Parking available

bath : Number of bathrooms available

carpet\_area : The carpet area of the Property

price\_in\_Lac : Price of the Property in Lakhs

Below variable are Categorical variable

House-type : Weather it’s a Flat or House

status : Ready to move in or under construction

transaction: Resale or New flat

furnishing : Furnish or Unfurnished.

ownership : Freehold or Cooperative society

facing : Which direction, East, West, North or East the property is facing

The downloaded data is located in the following file Incl\_Catg\_Var\_Data\_For\_NeuralNetwork.csv

**Format of the Data.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **location** | **bedroom** | **housetype** | **carpet\_area** | **status** | **transatcion** | **furnishing** | **balcony** | **park\_count** | **bath** | **ownership** | **facing** | **price\_in\_Lac** |
| Wagholi | 1 | Flat | 550 | Ready to Move | Resale | Unfurnished | 1 | 1 | 1 | Freehold | West | 26 |
| Wagholi | 1 | Flat | 560 | Ready to Move | Resale | Unfurnished | 1 | 1 | 1 | Cooperative Society | NA | 25 |
| Sinhagad Road | 3 | House | 2308 | Ready to Move | Resale | Unfurnished | 3 | 3 | 3 | Freehold | East | 175 |

**Predictor Development**

We have taken two approaches towards development of the predictor tools –

1. Python code for artificial neural network predictor
2. Excel based tool for artificial neural network predictor

**Python code for ANN predictor**

We have used the Google Colaboratory to developing the code for the predictor which internally uses predefined libraries viz. Pandas, Sklearn, Keras and Tensorflow in the google colaboratory environment.

The input data contains only numerical variables since the Neural Network model works with only numerical data.

For each of the feature variables it would have to be scaled to values between 0 and 1

After the scaling is performed, the value has to be split into Training, Validation and testing data.

In our case for the 1183 records, we have split it into 70%, 15% and 15% respectively.

After the data is split we have to create a Neural Network model. This is done in Python and the code is available in **realestate\_neuralnetwork.py**

We have also attached Jupyter Notebook file **RealEstate\_NeuralNetwork.ipynb .** Upload this file at [**https://colab.research.google.com/**](https://colab.research.google.com/) and it can be readily executed.

Number of Input Features used in the inputs data is 5.

Number of Hidden Layer is 1

Number of nodes in the layer is 13

Number of Output nodes = 1

Activation function in the input and Hidden layer is Rectified Linear Unit (ReLU)

Loss function is loss='mean\_squared\_error' and optimizer used is ‘adam’ optimizer.

The training over the input data is for 100 epochs in batches of 50.

Below is a sample of the execution with the original, Predicted price and absolute error

Actual-Price=[825.], Predicted-Price=971.87683, Error=[146.87683105]

Actual-Price=[32.], Predicted-Price=70.64985, Error=[38.64984894]

Actual-Price=[37.], Predicted-Price=36.926525, Error=[0.07347488]

Actual-Price=[105.], Predicted-Price=97.3235, Error=[7.67649841]

Actual-Price=[380.], Predicted-Price=1352.3433, Error=[972.34326172]

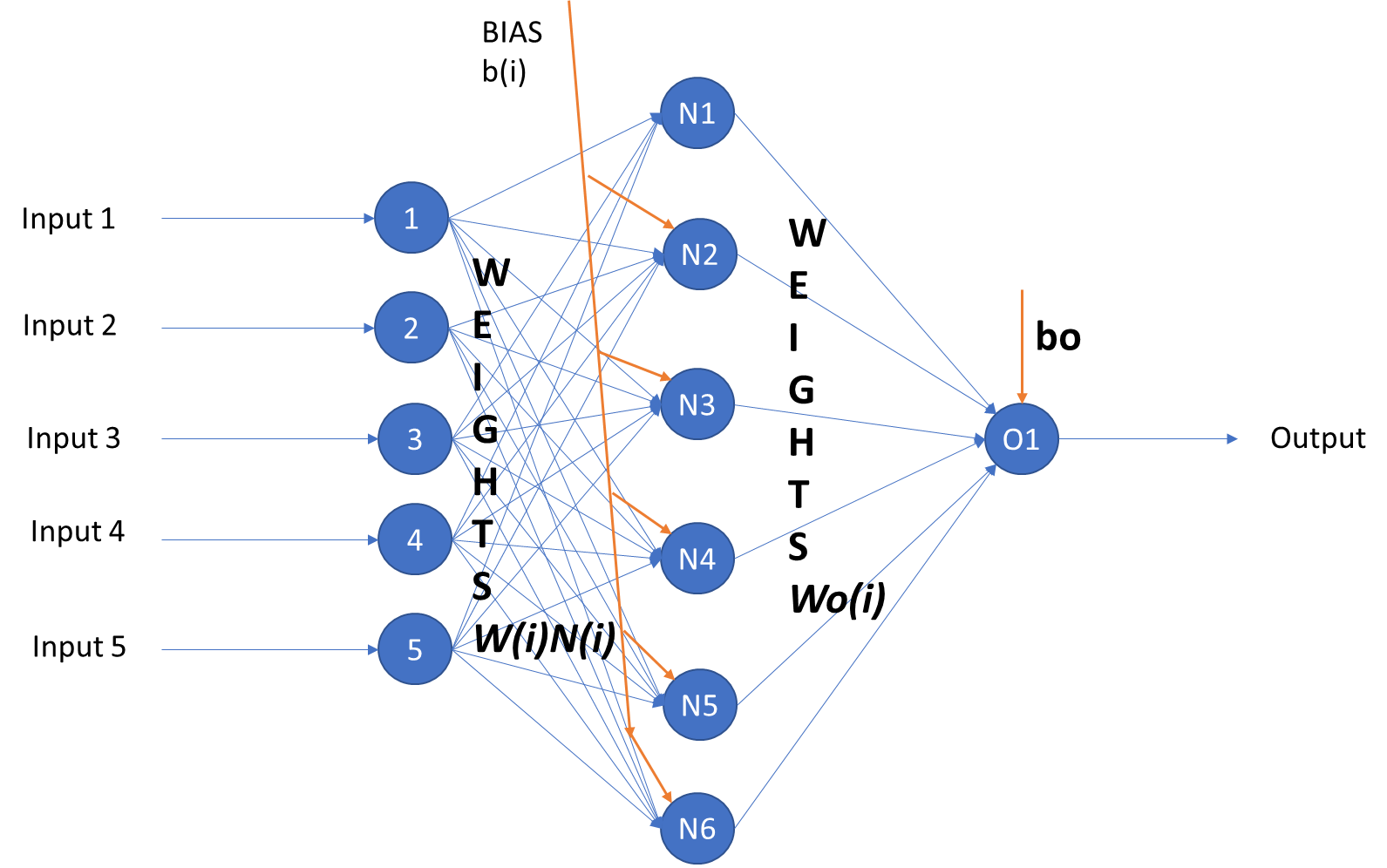
Actual-Price=[44.], Predicted-Price=83.05364, Error=[39.05364227]

Actual-Price=[33.], Predicted-Price=38.950558, Error=[5.95055771]

Actual-Price=[72.], Predicted-Price=99.98044, Error=[27.98043823]

**Excel based tool for ANN predictor**

To build an excel based ANN predictor, we have considered 5 inputs that is part of the data that has been downloaded from MagicBricks site. To structure the ANN we have consider a 5 inputs and 1 output. We have considered a single hidden layer with 6 neurons which is shown as below -



We have used the sigmoid function as the activation function and the objective while setting up the ANN is to minimize the sum of errors. The total dataset that has been divided in two parts and we have used initial 500 data points to train the ANN and another 500 data points to validate the ANN predictor. The results of the predictor are as below.

Weights matrix

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | W1 | W2 | W3 | W4 | W5 | b | Wo | bo |
| 1 | -8.85354 | -2.90255 | 0.04286 | 7.935268 | 0.017493 | 6.7459 | 0.29864329 | 0.4094445 |
| 2 | 0.38119 | 0.838974 | 0.42654 | 0.73643 | 0.600986 | 0.878195 | 0.00038067 | 0.55890671 |
| 3 | 1.083559 | 14.48796 | 0.004031 | 0.794604 | 0.012279 | 5.045473 | 0.47622896 | 0.68571561 |
| 4 | -6.96898 | -1.55657 | 0.011226 | 6.07995 | -0.02363 | 5.629336 | 0.13608681 | 1 |
| 5 | -9.0075 | -2.94216 | 0.047008 | 7.996171 | 0.014566 | 6.751433 | 0.30254117 | 0.6014114 |
| 6 | 1.272654 | 5.840205 | 0.006912 | 5.176208 | 0.14978 | -1.31135 | -0.3550765 | 0.49064456 |

Using this model to predict the estate price below are the results of the actual price Vs the predicted price

If you see the graph above we can see the predictor is able to predict with good accuracy, with certainly some outliers which is also clear from below chart

Based on the above graph we can see there is a consistent increase in predicted price with carpet area which is expected. However we can see the data is more scattered and some prices (actual) can be considered as outliers and might have to narrow down to a specific area within a city rather than considering the while city which we did in this case.

**Goodness of Predictor**

dd

**Conclusion**

Ddd

**References**

https://www.youtube.com/watch?v=vSzou5zRwNQ&feature=youtu.be and <https://randerson112358.medium.com/predict-house-median-prices-5f1a768dd256>

Code developed based on information at <https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/>

Reference : <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>

<https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/#:~:text=Adam%20is%20a%20replacement%20optimization,sparse%20gradients%20on%20noisy%20problems>.

<https://industryoutlook.cmie.com/>