House Price Prediction Using Neural Networks

Atharva Joshi
Machine Learning Intern
AI Technology and Systems
atharvajoshi10@gmail.com

***Abstract*—House prices increase every year, so there is a need for a system to predict house prices in the future. House price prediction can help the developer determine the selling price of a house and can help the customer to arrange the right time to purchase a house. The principal idea of this article is to predict the housing prices by building neural networks with three and five layers and compare the percentage error of the same. This report provides a way to perform regression using neural networks on Kaggle House Price Dataset. The report also shows a way of visualizing the data to remove the outliers.**

Keywords – Deep-Learning, Neural Networks, Machine-Learning, Regression

I. INTRODUCTION

Real Estate Property is not only the basic need of a man but today it also represents the riches and prestige of a person. Investment in real estate generally seems to be profitable because their property values do not decline rapidly. Changes in the real estate price can affect various household investors, bankers, policy makers and many. Investment in real estate sector seems to be an attractive choice for the investments. Thus, predicting the real estate value is an important economic index. India ranks second in the world in number of households according to 2011 census with a number of 24.67 crore. India is also the fastest growing major economy ahead of China with former's growth rate as 7% this year and predicted to be 7.2% in the next year. According to the 2017 version of Emerging Trends in Real Estate Asia Pacific, Mumbai and Bangalore are the top-ranked cities for investment and development. These cities have supplanted Tokyo and Sydney. The house prices of 22 cities out of 26 dropped in the quarter from April to June when compared to the quarter January to March according to National Housing Bank's Residex(residential index). With the introduction of Real Estate Regulation Development Act (RERA) and Benami property Act throughout the country India, more number of investors are attracted to invest into real estate in India. The strengthening and modernizing of the Indian economy has made India as attractive Investment destination. However, past recessions show that real estate prices cannot necessarily grow. Prices of the real estate property are related to the economic conditions of the state [2]. Despite this, we are not having proper standardized ways to measure the real estate property values.

II. RELATED WORK

In last two decades forecasting the property value has become an important field. Rise in the demand for property and unpredictable behaviour of economy compel researchers to find out a way that predict the real estate prices without any biases. Thus, it is a challenge for researchers to find out all the minute factors that can affect the cost of property and make a predictive model by taking into consideration all the factors. Building a predictive model for real estate price valuation requires a thorough knowledge on the subject.

The authors in [3] has scraped the housing dataset from Kaggle.com. The authors first pre-processed the data by removing the outliers. The authors used 4 regression techniques to predict the price value of the property. The four regression techniques used were : Ridge technique, Lasso Technique, Gradient Boosting and Hybrid regression. The Hybrid regression algorithm consisted of Lasso and Gradient boosting algorithms which provided better results and reduced the percentage error.

Using the same dataset, the authors in [2] used various machine-learning algorithms to predict the house prices. Algorithms such as Logistic Regression, SVM regressor, Lasso regression, Decision tree regression were used and their root-mean-squared errors (RMSE) were compared.

The authors in [1] scraped the dataset based on NJOP from Land and Building Tax (PBB) payment structure. The dataset contained 9 houses data in time series scattered in Malang City area, within 2014-2017. Normalization of data was done by completing the empty data at a certain time with the assumption that land prices tend to change every 2 years, while building prices tend to be stable. Using regression analysis and particle swarm optimization the root-mean-squared error was drastically reduced.

The author in [4] has compared hedonic price model and ANN model that predict the house prices. Hedonic price models are basically used to calculate the price of any commodity that are dependent on internal characteristics as well as external characteristics. The hedonic model basically involves regression technique that considers various parameters such as area of the property, age, number of bedrooms and so on. The Neural Network is trained initially and the weights and biases of the edges and nodes respectively are considered using trial and error method

III. METHODOLOGY

Methodology represents a description about the framework that is undertaken. It consists of various milestones that need to be achieved in order to fulfill the objective. I have undertaken different data mining and machine learning concepts.

A. Data Collection

The dataset used in this project is an open-source dataset taken from Kaggle.com. It consists of 3000 records with 80 parameters that have the possibility of affecting the property prices. Some of the parameters are Area in square meters, Overall quality which rates the overall condition and finishing of the house, Location, Year in which house was built, Numbers of Bedrooms and bathrooms, Garage area and number of cars that can fit in

garage, swimming pool area, selling year of the house and Price at which house is sold. The SalePrice is the label which we have to predict through regression techniques. Some parameters had numerical values while some had categorical values. Following are some of the parameters :

Parameters	Description	Datatype
SalePrice	The property's Sale Price in dollars.	Numerical
MSSubClass	The building class	Categorical
LotArea	Lot size in square feet	Numerical
Street	Type of road access	Categorical
GrLivArea	Above grade living area square feet	Numerical
GarageCars	Size of Garage in car capacity	Numerical
YrSold	Year Sold	Numerical
BldgType	Type of dwelling	Categorical
RoofStyle	Type of roof	Categorical
PoolArea	Pool area in square feet	Numerical

B. *Data Preprocessing*

It is a process of transforming the raw, complex data into systematic understandable knowledge. It involves the process of finding out missing and redundant data in the dataset. Entire dataset is checked for NaN and whichever observation consists of NaN will be deleted. Thus, this brings uniformity in the dataset.

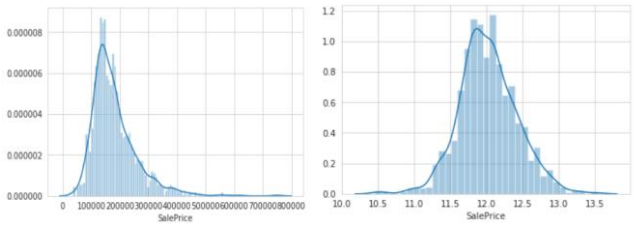
In this dataset there were numerous NaN values in numerical as well categorical columns. There were no redundant values in the dataset. The NaN values in the numerical columns were replaced with the mean of the values of the particular column whereas the NaN values in the categorical columns were replaced by simply a string 'None' so that each 'None' value can be encoded for further processing.

After filling up the NaN positions, encoding of the categorical data is done. Encoding refers to conversion of categories into numerical values. This was done using LabelEncoder of scikit-learn package. In the testing dataset, there were some values which were not seen in the training dataset. So the LabelEncoder was first fitted to combined dataset consisting of all the training and testing categorical samples and then the training and testing datasets were transformed. After the transformation, OneHotEncoding was done on the training and testing datasets. A one hot encoding is a representation of categorical variables as binary vectors. This was done to ensure that none of the categorical values that were converted to numerical values have a higher priority which could lead to assuming that the category is better than all the other categories. The OneHotEncoder of scikit-learn was used to perform the one hot encoding.

C. *Data Analysis*

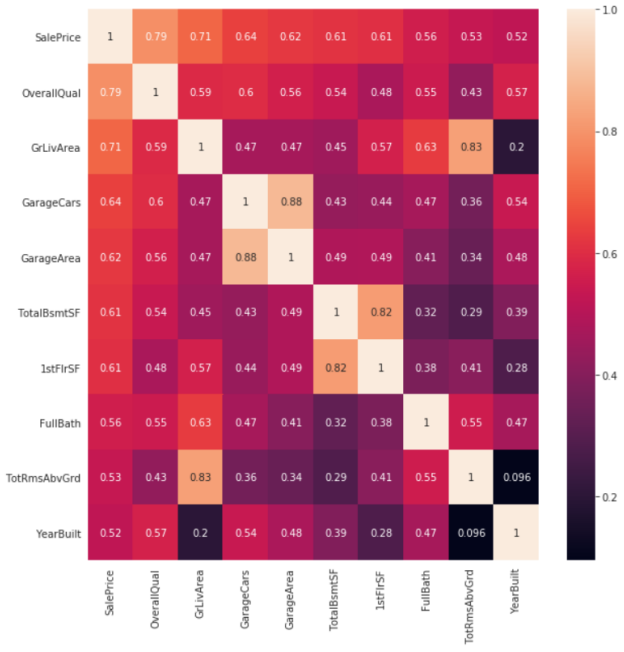
Before applying any model to our dataset, we need to find out characteristics of our dataset. Thus, we need to analyze our dataset and study the different parameters and relationship between these parameters. We can also find out the outliers present in our dataset. Outliers occur due to some kind of experimental errors and they need to be excluded from the dataset.

First of all, the label ie 'SalePrice' was visualized in order to find its distribution. Log transformation, in order to approximate the normal distribution, was applied to 'SalePrice'.



SalePrice without Log transformation SalePrice with Log Transformation

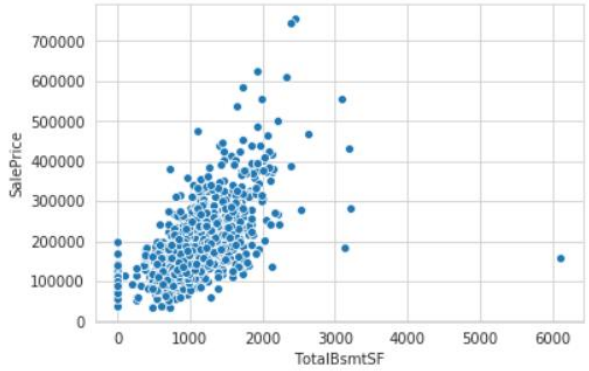
After analysing the label, a Heatmap was generated in order to analyse the correlation between the parameters. A heatmap is a graphical representation of data that uses a system of color-coding to represent different values. The heatmap is plotted using a correlation matrix. Correlation Matrix gives a in depth idea about correlation among various parameters. A correlation number gives the degree of association between two variables. The correlation number exists between +1 to -1. A positive number represents a positive correlation between two variables and vice versa.

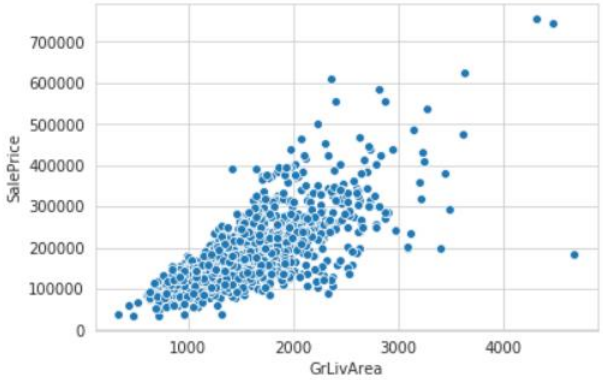


Heatmap for maximum correlating parameters with respect to 'SalePrice'

From the heatmap, we can deduce that 'GrLivArea', 'TotalBsmtSF' and 'OverallQual' are the most correlated with 'SalePrice'.

We can analyse the columns that are most correlated to 'SalePrice' by using a Pairplot. A Pairplot allows us to see both distribution of single variables and relationships between two variables . It was deduced that 'TotalBsmtSF' and 'GrLivArea' had some outliers. To remove the outliers scatterplots were plotted in order to find the index positions of the outliers.





The outliers resided in the index position where 'TotalBsmtSF'>6000 and ('GrLivArea'>4000 and 'SalePrice' < 20000). These outliers were removed by dropping the corresponding rows.

D. Training

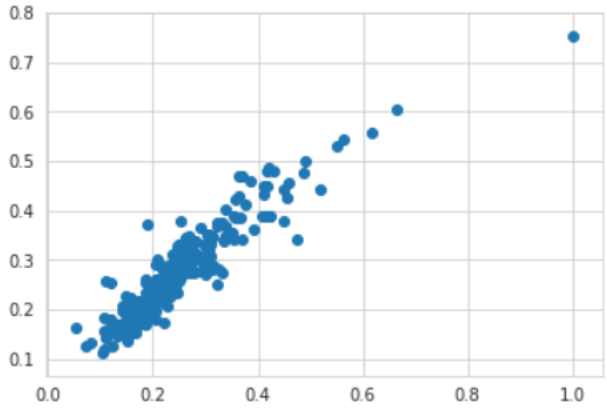
Once the data is clean and we have gained insights about the dataset, we can build an appropriate neural network that fits our dataset. As the label in our dataset is continuous, we need to fit a regression model to predict the House prices. Regression is a technique used to model and analyze the relationships between variables and often times how they contribute and are related to producing a particular outcome together. A linear regression refers to a regression model that is completely made up of linear variables.
The training set was divided in training and validation datasets in order to test the model and minimize the mean absolute percentage error. The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation(MAPD), is a measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation, also used as a loss function for regression problems in machine learning.

1. Artificial Neural Network with 3 layers

A neural network with 3 layers was built using keras – a python API which runs on top of tensorflow. The neural network consisted of 259 input neurons (259 due to the addition of columns using OneHotEncoder) and 128 output neurons. It consisted of 1 hidden layers with 128 output neurons and 1 output layer with 1 neuron. The activation function used was 'relu' to introduce non-linearity in the output. For the output layer, the activation function used was 'linear'. For the updation of weights, 'adam' algorithm was used with mean squared error as the loss function. To tackle the problem of overfitting, dropout layers were added with a dropout rate of 0.15. The model was trained for 15 epochs with batch size of 25. It was found that after 15 epochs, the model overfitted the data which resulted in high mean absolute percentage error. To improve the performance of the model, GridSearchCV was used to test various hyper parameters in order to select the best parameters. The model was tested for:

a) Batch sizes: 20,25,30
b) Epochs: 5,10
c) optimizers: adam and rmsprop

it was observed that for batch size 20, epochs 5 and adam optimizer, the model performed best with an error of 13.52%.



Scatter plot of predicted values using 3 layer Neural Network

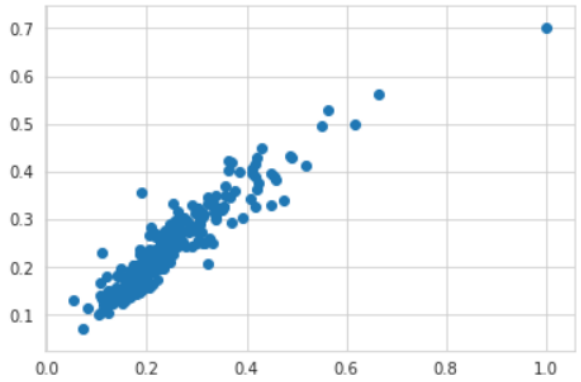
2. Artificial Neural Network with 5 layers

A neural network with 5 layers was built using keras – a python API which runs on top of tensorflow. The neural network consisted of 259 input neurons (259 due to the addition of columns using OneHotEncoder) and 128 output neurons. It consisted of 3 hidden layers with 128 output neurons and 1 output layer with 1 neuron. The activation function used was 'relu' to introduce non-linearity in the output. For the output layer, the activation function used was 'linear'. For the updation of weights, 'adam' algorithm was used with mean squared error as the loss function. To tackle the problem of overfitting, dropout layers were added with a dropout rate of 0.05. The model was trained for 15 epochs with batch size of 25. It was found that after 15 epochs, the model overfitted the data which resulted in high

mean absolute percentage error. To improve the performance of the model, GridSearchCV was used to test various hyper parameters in order to select the best parameters. The model was tested for:

a) Batch sizes: 20,25,30
b) Epochs: 5,10
c) optimizers: adam and rmsprop

it was observed that for batch size 20, epochs 5 and adam optimizer, the model performed best with an error of 11.2%.



Scatter plot of predicted values using 5 layer Neural Network

IV. CONCLUSION

In this report, I have built two neural neworks to predict the house prices. I have mentioned step by step procedure to analyze the dataset and finding the correlation between the parameters. These feature set was given to the two models and a csv file was generated consisting of predicted house prices. It was concluded that with the right amount of dropout rate, the 5 layer neural network can outperform the 3 layer neural network with an approximate error of 2%. The 5 layer Neural Network can easily overfit the data, so the right amount of epochs and batch size were found out using GridSearchCV. After careful evaluation of the models, it was deduced that the number of output neurons for the input layer and the hidden layers must be the average of variables in the training and testing dataset. For future work, other preprocessing techniques can be used and the training dataset can be clustered by taking into account features such as 'Neighbourhood' which will help to improve the error.

V. REFERENCES

[1] Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, wayan Firdaus Mahmudy, "Modeling House Price Prediction using Regression Analysis and Particle Swarm Optimization", Vol. 8, No. 10, 2017

[2] Neelam Shinde, Kiran Gawande, "Valuation Of House Prices using Predictive Techniques", Volume-5, Issue-6, Jun.2018

[3] Sifei Lu, Zengxiang Li, Zheng Qin, Xulei Yang, Rick Siow Mong Goh ,"A Hybrid Regression Technique for House Prices Prediction",December-2017

[4] Limsombunchai, Visit. "House price prediction: hedonic price model vs. artificial neural network."New Zealand Agricultural and Resource Economics Society Conference. 2004.