HOUSE PRICE PREDICTION USING NEURAL NETWORKS

Gaurav Arora
Machine Learning Intern
AI Technology and Systems
garora039@gmail.com
www.ai-techsystems.com

***Abstract* - Housing prices are an important reflection of the economy, and housing price ranges are of great interest for both buyers and sellers. In this project, house prices will be predicted given explanatory variables that cover many aspects of residential houses. The goal of this project is to create a regression model that are able to accurately estimate the price of the house given the features. The heart of the problem lies within to develop an efficient 3 layer and a 5 layer neural network architecture that can make house prices predictions and to compare there percentage errors.**

Keywords - exploratory data analysis, feature engineering, regression analysis, tableau visualization, neural networks, keras, deep learning

I. INTRODUCTION

A country's economic performance has direct repercussions on the dynamics of various markets, especially real estate. Over the last few years, various economic issues have declined the overall demand, specifically in the residential sector . Houses is not only the basic need of a man but today it also represents the riches and prestige of a person. real estate is a complex and an opaque market, which depends on several price factors like interest rates, rental yield and transaction volumes etc. Demand in general is one of the key factors in increase or decrease of the prices of any commodity. The property demand depends on positive economic growth, job and income prospects and lower property prices. The demand is high due to strong population growth, rise in nuclear families, continuing urbanization trends and improved regulatory framework. In housing real estate, demand for a particular area is inversely proportional to its supply. Limited supply in housing real estate causes the prices to increase. Conversely an oversupply leads to a decrease in the prices. According to the studies, in the year 2014, a sharp decline of about 30% was observed in demand in the seven major cities in India. This is mainly attributed to high prices, higher interest rates and cautious buyer sentiments. The developers responded to the decrease in demand by reducing the supply whereby there was a 25% decline on a year on year basis. The decline was reported in the premium and high end/mid end business segments, was observed across all the major cities, steepest in the NCR.

II. RELATED WORK

The real estate market is exposed to many fluctuations in prices because of existing correlations with many variables, some of which cannot be controlled or might even be unknown. Housing prices can increase rapidly (or in some cases, also drop very fast), yet the numerous listings available online where houses are sold or rented are not likely to be updated that often. In some cases, individuals interested in selling a house (or apartment) might include it in some online listing, and forget about updating the price. In other cases, some individuals might be interested in deliberately setting a price below the market price in order to sell the home faster, for various reasons.

A. ***Real Estate oppurtunities using Machine Learning***

The authors in [2] aimed at developing a machine learning application that identifies opportunities in the real estate market in real time, i.e., houses that are listed with a price substantially below the market price. This program can be useful for investors interested in the housing market. The application is formally implemented as a regression problem that tries to estimate the market price of a house given features retrieved from public online listings. For building this application, they have performed a feature engineering stage in order to discover relevant features that allows for attaining a high predictive performance. Several machine learning algorithms have been tested, including regression trees, k-nearest Neighbors, support vector machines and neural networks, identifying advantages and handicaps of each of them.

B. ***Hedonic Pricing***

The author in [1] proposed an hedonic model for determining the willingness of house buyers to pay for clean air. An hedonic model is a model that decomposes the price of an item into separate components that determine its price. For example, an hedonic model for the price of a house may decompose its price into the house characteristics, the kind of neighborhood, and the location

C. ***Particle Swarm optimization***

The authors in [3] worked on predicting the house prices based on NJOP houses in Malang city with regression analysis and particle Swarm optimization(PSO). PSO is used for selection of affect variables and regression analysis is used to determine the optimal coefficient in prediction. The result from this research proved combination regression and PSO is suitable and get the minimum prediction error obtained which is IDR 14.186.

III. METHODOLOGY

A. ***Dataset***

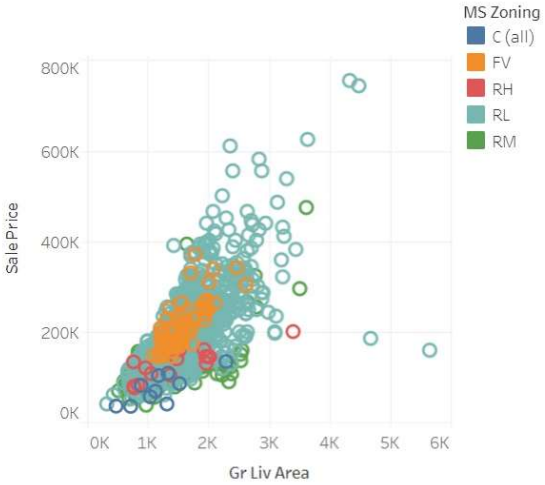
This Ames Housing dataset was compiled by Dean De Cock for use in data science education. It contains almost 79 explanatory features describing every aspect of residential homes in Ames. The dataset contains both numerical and categorical features. It has about 3000 data points. Some of the parameters are year when the house was built, General shape of the property, number of bedrooms and cars, Neighborhood, type of roof, overall quality and condition of the house, garage type and Sale condition. The SalePrice is the target variable which we have to predict using regression.
Some of the parameters are listed below.

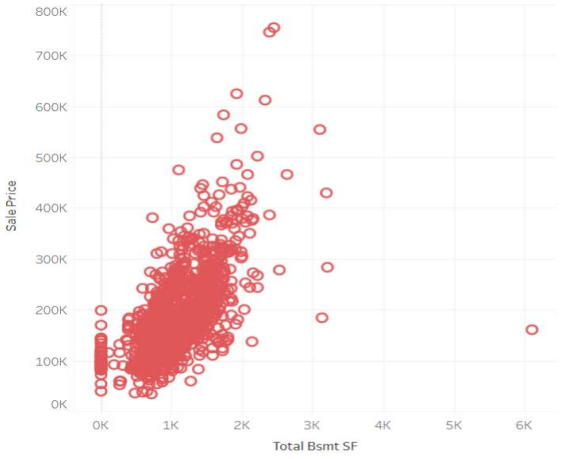
Parameters	Description	Datatype
SalePrice	The property's Sale Price in dollars.	Numerical
MSSubClass	The building class	Categorical
LotArea	Lot size in square feet	Numerical
Street	Type of road access	Categorical
GrLivArea	Above grade living area square feet	Numerical
GarageCars	Size of Garage in car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

B. ***Exploratory Data Analysis***

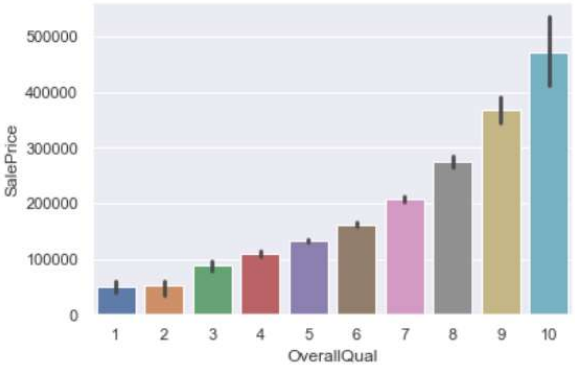
Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

We performed some bivariate analysis on the data to get a better overview of the data and to find outliers in our dataset. Outliers can occur due to some kind of errors while collecting the data and need to be removed so that it don't affect the performance of our model.

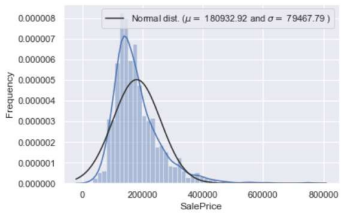


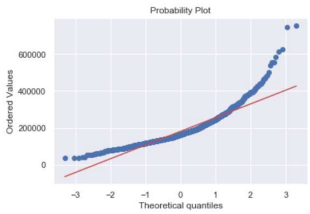


While performing data analysis we notice that the feature OverallQual plays a huge role in deciding the price of a house.



Then the target variable SalePrice was visualized in order to find its distribution. Then Log Transformation was applied on it to make it a normal distribution.

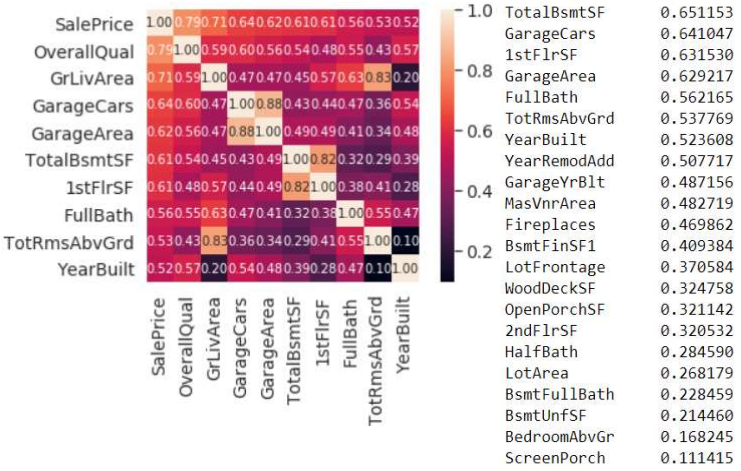




After this, we found the most important features relative to the target by building a correlation matrix. A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. The correlation coefficient has values between -1 to 1

1. A value closer to 0 implies weaker correlation (exact 0 implying no correlation)
2. A value closer to 1 implies stronger positive correlation
3. A value closer to -1 implies stronger negative correlation.

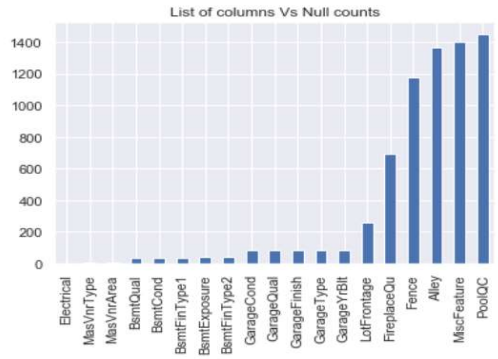
Most important features -



C. *Data Preprocessing*

Many real-world datasets may contain missing values for various reasons. They are often encoded as NaNs, blanks or any other placeholders. Training a model with a dataset that has a lot of missing values can drastically impact the machine learning model's quality. Some algorithms such as *scikit-learn estimators* assume that all values are numerical and have and hold meaningful value. One way to handle this problem is to get rid of the

observations that have missing data. However, you will risk losing data points with valuable information. A better strategy would be to impute the missing values.
We deleted all those columns from the dataset which were having more than 40% null values in them as these columns were not giving us any significant information.



After that we calculated the missing ratio of each column and then filled the missing values with basic imputation methods. Basic imputation methods include filling the null point with mean, median or mode of the data.

	Missing Ratio
FireplaceQu	47.325103
LotFrontage	17.764060
GarageCond	5.555556
GarageQual	5.555556
GarageFinish	5.555556
GarageYrBlt	5.555556
GarageType	5.555556
BsmtFinType2	2.606310
BsmtExposure	2.606310
BsmtFinType1	2.537723
BsmtCond	2.537723
BsmtQual	2.537723
MasVnrArea	0.548697
MasVnrType	0.548697
Electrical	0.068587

The dataset consists of features in various formats. It has numerical data such as prices and numbers of bathrooms/bedrooms/living rooms, as well as categorical features such as zone classifications for sale, which can be 'Agricultural', 'Residential High Density', 'Residential Low Density', 'Residential Low Density Park', etc. In order to make this data with different format usable for our algorithms, categorical data was converted into separated indicator data, which expands the number of features in this dataset. There were 37 categorical and 36 numerical columns. After encoding these 73 columns now our dataset had 335 columns.

D. ***Model***

Once the data is cleaned we will now proceed further to make our neural network model. As our target variable is continuous we will fit a regression model to the dataset. The SalePrice is in dollars and we will try to predict it using a 3 layer and a 5 layer neural network.

A neural network is a computational system tahat creates predictions based on existing data.

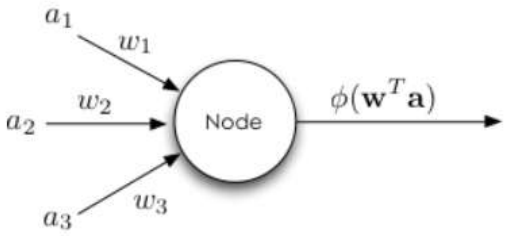
A neural network consists of :

Input layers : Layers that take inputs based on existing data

Hidden layers : Layers that use backpropogation to optimize the weights of the input variables in order to improve the predictive power of the model.

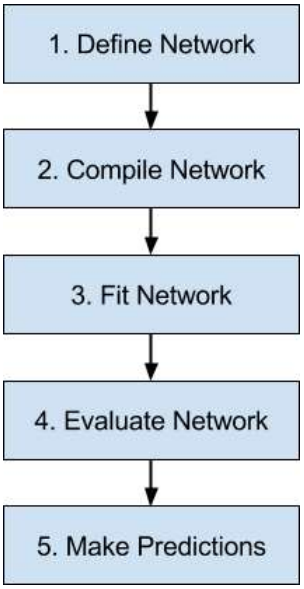
Output layers : output of predictions based on the data from input and hidden layers.

The each node receives a set of weighted inputs, process their sum with its activation function, and passes the result of the activation function to nodes further down the graph.



There are several canonical activation functions like linear, sigmoid, tanh, ReLu , leaky ReLu

There are 5 steps in the neural network model life-cycle -



1. ***Neural network with 3 layers***

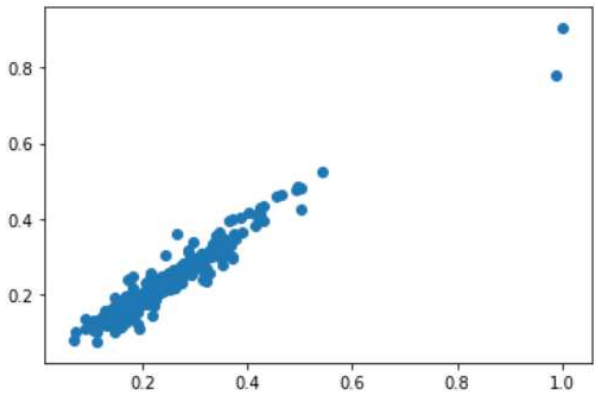
We will build neural networks using Keras which is an open-source library written in python. It is capable of running on top of TensorFlow and Theano. The neural network consist of 3 layers. The input layer has 150 nodes and excepts row of data with 335 variables(input_dim = 335 argument). The first hidden layer has 75 nodes and uses the relu activation function. The output layer has only one node because we are trying to predict the house prices which is a regression problem and uses linear activation functions. Now that we have defined the model, we can compile it. When compiling, we must specify some additional properties required when training the network . We must specify the loss function to use to evaluate a set of weights, the optimizer is used to search through different weights for the network and any optional metrics we would like to collect and report during training. In this case we use adam algorithm as the optimizer and Mean Squared Error as the loss function. In this model 2 dropouts layers were also added to reduce over-fitting. Dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 150)               50400
_________________________________________________________________
dropout_3 (Dropout)          (None, 150)               0
_________________________________________________________________
dense_7 (Dense)              (None, 75)                11325
_________________________________________________________________
dropout_4 (Dropout)          (None, 75)                0
_________________________________________________________________
dense_8 (Dense)              (None, 1)                 76
=================================================================
Total params: 61,801
Trainable params: 61,801
Non-trainable params: 0
```

Model Summary :

Epochs = 100
Batch size = 32
Dropout percentage = 5 %
No. of layers = 3
Loss function = Mean Squared Error
Optimizer = Adam

Then the model was fitted to the train data and it was seen that with the above parameters model performs best with a mean absolute percentage error of 3.8897 % and a MSE of 0.000837020



Scatter plot of predicted values (3 layer neural network)

II. ***Neural network with 5 layers***

We will build neural networks using Keras which is an open-source library written in python. It is capable of running on top of TensorFlow and Theano. The neural network consist of 5 layers. The input layer has 256 nodes and excepts row of data with 335 variables(input_dim = 335

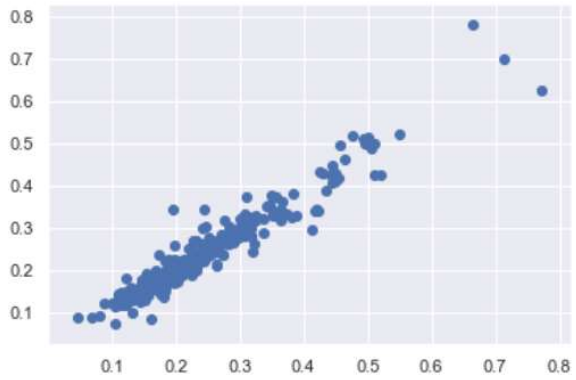
argument). All the 3 hidden layer have 128 nodes each and uses the relu activation function. The output layer has only one node because we are trying to predict the house prices which is a regression problem and uses linear activation functions. Now that we have defined the model, we can compile it. In this case we use adam algorithm as the optimizer and Mean Squared Error as the loss function. In this model 2 dropouts layers were also added to reduce over-fitting. All the layers uses a uniform Kernel intializer to intialize the value of weights.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	86016
dense_2 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 128)	16512
dropout_2 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 128)	16512
dense_5 (Dense)	(None, 1)	129

Total params: 152,065
Trainable params: 152,065
Non-trainable params: 0

Model Summary :

Epochs = 100
Batch size = 32
Dropout percentage = 5 %
No. of layers = 5
Loss function = Mean Squared Error
Optimizer = Adam



Scatter plot of predicted values (5 layer neural network)

Then the model was fitted to the train data and it was observed that with the above parameters model performs best with a mean absolute percentage error of 3.4514 % and a MSE of 0.000888634.

IV. CONCLUSION

In this project we tried to predict the house prices using two neural networks – 3 layer and a 5 layer. First I analysed the data and observed the trends in it and got rid of the outliers. Then I preprocessed the data and handled null or missing values. The main step in this case study was the feature engineering, identifying which features are most correlated with the target variable and which feeatures are least correlated. These neural networks can easily overfit so keeping that in mind we added Dropout layers in the model. With the right value of hyperparameters – batch size, epochs and dropout percentage, the 5 layer neural network can perform better than a 3 layer neural network. For future work more advanced imputation methods can be implemented like k-Nearest neighbor(kNN) Imputation and Multiple Imputation by Chained Equations which works with the assumption that the missing data are Missing at Random(MAR).

V. REFERENCES

[1] Harrison, D., and D. L. Rubinfeld. 1978. “Hedonic Housing Prices and the Demand for Clean Air.”*J. Environ. Econ. Manag.*5 (1): 81–102.

[2] Alejandro Baldominos , Iván Blanco , Antonio José Moreno , Rubén Iturrarte , Óscar Bernárdez and Carlos Afonso, “Identifying Real Estate Oppurtunities Using Machine Learning, November 2018

[3] Adyan Nur Alfiyatin, Hilman Taufiq, Ruth Ema Febrita, Wayan Firdaus Mahmudy, “Modeling House Price Prediction using Regression and Particle Swarm Optimization”, Vol. 8, No. 10,2017