Background pattern

Description automatically generated with low confidence

**SECOND ASSIGNMENT**

**Developing a Neural Network for solving a Regression problem**

Rahul Kumar

# Table of Contents

[Table of Contents 2](#_Toc100575454)

[Abstract/Executive Summary 3](#_Toc100575455)

[Introduction 3](#_Toc100575456)

[Data Cleaning 4](#_Toc100575457)

[Data Exploration 4](#_Toc100575458)

[Evaluation 5](#_Toc100575459)

[Conclusion 9](#_Toc100575460)

[References 9](#_Toc100575461)

# Abstract/Executive Summary

This report aims to investigate house sale prices for king country by developing a neural network. The dataset consists of 21613 rows and 21 columns. The neural network develop uses 16 different features to predict the price of the house. The data collected is based on characteristics of the housing market that would affect the price of a house such as size, number of rooms, condition and zip code within king country. The report concludes that house prices are closely associated with sqrt\_living15, grade, sqft\_above, number of bathrooms, views, year of renovation and number of times, how much old the house is and the number of times the house got renovated. This means that the more space available to a property the higher its price will be. It also shows that newer houses also have a higher price this is expected due to market preferences.

# Introduction

House pricing is an important factor within the economy as it is linked to consumer spending. If house prices are high, it gives homeowners confidence and increases their wealth. This allows them to take larger loans against the value of the home to spend further within the economy on supplementing their pensions, renovating their homes, spend more on goods and services or pay off other debts. Therefore, it is important to know the value of properties within different areas to distinguish the current economic prosperity of the area.

This report will look at the various characteristics of each house within the area and how this analysis can be used to value the properties within the area king country.

We have used google colab which is a cloud-based infrastructure for coding this Assignment. There are several libraries are used to complete this assignment. Retrieving and Preparing of the data is done using pandas and numpy Library. Data Exploration or Exploration analysis of the data is done using the seaborn and matplotlib libraries. Splitting the data into training and testing and feature scaling is done under data modelling section using sklearn library. Finally, we used keras and tensorflow2(RandomizedSearchCV) to train our neural network and tune different hyperparameter to find an optimal neural network.

To test different hyperparameter of the neural network we selected on hyperparameter kept others constant and trained the neural network using different value of the selected hyperparameter using RandomizedSearchCV to find out the optimal value of the selected hyperparameter best on score, MSE, MAE and RMSE. The training and test data was split 30% and 70%. The impact of hyperparameter is evaluated based on score, MAE, MSE and RMSE.

The different hyperparameter and their value tested are listed below: -

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Hyperparameter** | **Values** |
| 1. | Learning rate | 0.1, 0.01, 0.001 and 0.0001. |
| 2. | Activation function | sigmoid, ReLU, Tanh, Elu, LeakyReLU and PReLU |
| 3. | Epoch | 25, 50, 75 and 100. |
| 4. | Number of neurons | 4, 8, 12 and 16 |
| 5. | Optimiser | RMSprop, Adagrad, Adadelta, Adam, Adamax and Nadam |
| 6. | Loss function | mean squared error and mean absolute error |
| 7. | Batch size | 32, 64 and 128 |

# Data Cleaning

There were no missing values in the data set. There are only integer and float data types. The id, date, lat, long and zip code columns were removed as our focus is to see the price of the house w.r.t to its features that directly impact the price of the house. We have created three more columns as a feature (age\_built, age\_renovated and done\_reno) columns from existing columns to calculate the year in which the house was build, how many it’s been since last renovation and to know if the house has been renovated or not.

# Data Exploration

Chart, histogram

Description automatically generatedIn this part we tried to explore different corelation different features and target.so that we can figure out which features are strongly corelated with target.

House price per house grade shows that as the grade of the house increases the value and price of the house also increases as the graph below shows.

Chart, scatter chart

Description automatically generated

The scatterplot of sqft\_living compared to price shows that as the sqft\_living increase, the price of the house also increases. It can also be observed that they are very few houses whose sqft\_living is greater than 10000.

Chart

Description automatically generated with low confidence

The target column(price) is shown in histogram, and it is positively skewed. We can see a lot of outliers as we expected. i.e., it has a very wide price range, but it doesn't mean the data is wrong here.

Chart

Description automatically generated

The heatmap shows price is closely associated with sqrt\_living. That grade, sqft\_above, sqrt\_living15, bathrooms, and views have high correlation which isn’t surprising because they are supposed to be highly correlated. The more space a property has, the higher the price of the property. The heatmap also shows that year built negatively correlates with age built and age renovated which makes sense giving that most homes being built increased over time showing that most homes are younger. This also has led to higher prices.

# Evaluation

We have generated baseline Neural Network using Sequential() model with 16 neurons since we have 16 features, learning rate as 0.1, batch size as 128, epoch as 50, activation function as relu, 16 hidden layers, optimizer as Adam and loss function as mean\_squared\_error. After training the baseline neural network we get below error score: -

|  |  |
| --- | --- |
| MAE: | 122991.66381745806 |
| MSE: | 35079678666.888824 |
| RMSE: | 187295.69847406753 |

Below graph shows the Model loss progression during entire training:

Chart

Description automatically generated

From above figure we can say that baseline neural network is overfitting since fits the data well and the model is failing to generalize.

Our overall strategy was to tune the hyperparameters using RandomizedSearchCV to optimize the neural network.

The following are the hyperparameters that can be tuned to optimize the classifier.

* Training length (number of epoch)
* Batch Size (how many rows to look at once during single training step).
* Activation Function
* Learning rate (how to fast to learn)
* Optimizer
* Loss Function
* Number of neurons

Note: We have kept cv = KFold (2) and verbose=5 for all hyperparameters tuning.

The Hyper parameter Tuning with cross validation using RandomizedSearchCV showed that for each hyperparameter:

Results after Tuning the learning Rate is shown below: -

Text

Description automatically generated

The neural network performs best at learning rate = 0.001

1. Best Score: -44631541760.0
2. MAE: 128452.3413245585
3. MSE: 38193256003.32965
4. RMSE: 195430.94945102645

Results after Tuning the Activation function is shown below: -

Text

Description automatically generated

The neural network performs best for PReLU activation function.

1. Best Score: -42834239488.0
2. MAE: 124182.36591935437
3. MSE: 33977091851.737972
4. RMSE: 184328.76024033246

Results after Tuning the epoch is shown below: -

Text

Description automatically generated

The neural network works best for epoch = 75.

1. Best Score: -45256589312.0
2. MAE: 132020.17285096005
3. MSE: 40880045492.82813
4. RMSE: 202188.14379885912

Results after Tuning the Number of Neurons is shown below: -

Text

Description automatically generated

The neural network performs best with 16 neurons, since there are 16 features in the dataset.

1. Best Score: -45969188864.0
2. MAE: 132978.17859390678
3. MSE: 41312707614.51695
4. RMSE: 203255.27696597928

Results after Tuning the Optimiser is shown below: -

Text

Description automatically generated

The neural network performs best Nadam optimizer.

1. Best Score: -44617385984.0
2. MAE: 127507.9685210711
3. MSE: 37178244893.11222
4. RMSE: 192816.60948453643

Results after Tuning the Loss Function is shown below: -

Text

Description automatically generated

The neural network performs best when loss function is mean\_absolute\_error

1. Best Score: -136193.6953125
2. MAE: 128463.02451685636
3. MSE: 45475642115.84365
4. RMSE: 213250.18667247082

Results after Tuning the Batch size is shown below: -

Text

Description automatically generated

The neural network performs best for batch\_size = 32

1. MAE: 130706.32520690249
2. MSE: 39905912150.538574
3. RMSE: 199764.64189274982

The plotted results of the activation function giving the best rsme is shown below in the scatterplot:

Chart, scatter chart

Description automatically generated

Form above result we can conclude that the Neural network performs better when number of neural networks is equal to the number of features which is 16, it also performs better when learning rate is low (0.001).

We often incline towards using higher batch size as it speeds up the computational process, but too large of a batch size might lead towards the poor generalisation. Here neural network gives best score 32 batch size. In our case neural network performs best when we use Nadam optimiser as it helps in reducing the overall loss and improve the accuracy. PReLU activation function, 75 epochs and when loss function is mean absolute error.

# Conclusion

The challenges of this project were cleansing and preparing the data as we had to remove columns as our focus was to see the price of the house w.r.t to its features. Calculate the age of the house and create more columns as features of other columns to eventually have a set of data to use. Choosing the initial parameters was tough because we wanted to ensure that we chose values that would best fit the scenario. One very large challenge was communicating the needs from coding to the report and relaying the required information. We overcame this by ensuring we maintained good communication throughout the project.

Rahul did the coding, analysis and prepared the readme file. James wrote the draft report, reviewed and finalised by Rahul and the two of us put the project together.

# References

1. [https://www.kaggle.com/datasets/harlfoxem/housesalesprediction/discussion/207885](https://www.kaggle.com/datasets/harlfoxem/housesalesprediction/discussion/207885%20)
2. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html>
3. <https://medium.com/geekculture/randomizedsearchcv-e6444c457c8d>
4. <https://www.tensorflow.org/api_docs/python/tf/keras/Sequential>
5. <https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html>
6. <https://matplotlib.org/3.5.0/api/_as_gen/matplotlib.pyplot.hist.html>
7. <https://matplotlib.org/3.5.0/api/_as_gen/matplotlib.pyplot.scatter.html>
8. <https://seaborn.pydata.org/tutorial/color_palettes.html>
9. [https://seaborn.pydata.org/generated/seaborn.heatmap.html](https://myoffice.accenture.com/personal/ruchita_agrawal_accenture_com/Documents/Downloads/Copy%20of%20Global%20Holiday%20Tracker_2022%20as%20published%20on%2030-March-22.xlsx?web=1)