MATH 404 Final Project
Housing Price Retrieval from Local Market Data

Andrew Hartman, Niko Hansen, Collin Heist, and Cory Holt

May 14, 2020

Contents

1 Problem Statement

The main motivation for this project is to create a network that can estimate the actual market price based on significant pricing parameters of sold houses in the same locality. Most sale prices rely on some form of algorithm created by the realtor. If potential home buyers could utilize their own algorithm, they could differentiate a fair price from an unfair one. Home buying has quite a bit of variability due to the relative instability of the housing market, and its large positive and negative swings. Thus, another problem that is identified in this problem is the actual sale of a home which can depend quite a bit on the condition of the housing market at the time of sale.

2 Dataset

The original dataset (see **Section 7 - References**) was collected from the individual sale of residential houses in *Ames, Iowa* from 2006 to 2010. There are a total of 2,930 observations with a total of 81 data points including the sales price. Using this dataset, it was trimmed down to only hold the columns that were the most influential and would work within the proposed method. This meant that opinionated quality and condition columns were removed, as well as columns that had 98-100% of the same value for each point.

After the removal of a fair bit of data we deemed unnecessary, the remaining data was able to be split into two distinct groups. There was categorical data; such as number of bedrooms or bathrooms, as well as continuous quantitative data such as square footage or lot area.

3 Proposed Method

The goal for this project is to train a network to predict housing prices based on a set of parameters input for your house. It will be trained using a dataset of close to 3,000 homes sold in *Ames, Iowa*. Once trained a potential user could input information about a prospective house in the area and it would provide a recommended price for the house based on the learned information of the area.

To train the network, we found that two distinct strategies existed. First, the network could be trained using data that is categorical; and second, the network could be trained using data that is purely quantitative. In order to continue this discussion, we will let the categorical approach be termed Proposal 1, and the quantitative approach will be termed Proposal 2.

For Proposal 1, the categorical approach involves separating the data into one-hotted vectors. For some data points such as *number of bathrooms*, or total *number of rooms*, this one-hotting is fairly intuitive and straightforward, as each data point would fall into a specific category. However, for parameters such as *lot area* whose values range in the thousands, one-hotting each value does not make sense as it is too specific a description to provide an accurate result. Instead, the data for these parameters must be binned so that each data point is one-hotted to a range of values rather than to a specific value. In this way, we also reduce the dimension of this input data by creating these ranges.

For Proposal 2, the quantitative approach involves feeding in the raw data and treating the model as more of a regression type model. The output for such a model would just be a single numerical value. As the value of homes may range quite a bit due to external factors that fall outside of the scope of this project, a price range would serve as a better estimate in order to take into account for the unseen factors that can influence home value. Therefore, we chose Proposal 1 as our best method to execute. We feel that a categorical approach to this project would work better to actually give a useful estimation for the price of the home. This is due to the range in price that the model would give, which accounts for the variability of a market value. Along the same lines, using a categorical one-hotted approach would provide a better representation of some of the larger numerical categories.

For example, for the category *square footage*, a home with slightly more square footage than another may not directly correlate to increased overall value, but a larger difference in square footage will most likely show a correlation. Therefore, by grouping similar values in the same range we would clearly define a boundary at which increased square footage impacts overall value, therefore, improving the correlation between the inputs and outputs.

4 Executed Method

As discussed in the section above, we chose the categorical approach for this project. The process for our network creation was split into four distinct phases: data preparation, network construction, and network training / tuning. Each phase is described below.

For the data preparation phase, we sorted through the redacted data and further redacted it down to the parameters that we deemed most influential and most significant in correlation to the overall value of the home. These parameters were:

- Lot area
- Year built
- Year remodeled
- Basement square footage
- First floor square footage
- Second floor square footage
- Gross living area
- Full baths
- Half baths
- Bedrooms above ground
- Total rooms above ground
- Garage cars
- Garage area
- Year sold

The labels for each data point is the *sale price*. The parameters whose values did not exceed more than a few dozen were easily one-hotted, while those with much larger total ranges were binned into one-hotted vectors. The binning involved walking through each data point in the given vector and checking which range of value or bin that data point fell into. This provided a logical vector for each data point which also served to one-hot the data. The *sale price* data was also included in this binning. To accomplish the binning, the Pandas data frame function **DataFrame.cut()** was used. This allowed us to automate the binning and very easily change the number of bins or adjust data distribution in bins to see if that would affect rates down the line.

The network architecture that we implemented was a Keras model with 4 dense layers, where three apply a Relu activation function and then the fourth applies a Softmax at the end to produce the predicted output. Each layer had a dimension of 32 except the Softmax which was 39, the size of our labels. This was ran with Categorical Crossentropy loss function and the Adam optimizer.

Most of the internal parameters such as learning rate and batch size are handled by Keras so they were not modified for testing. However, the number of nodes for each layer was changed and re-ran a few times to test what the best solution would be, having 32 nodes in each layer worked out the best. A big way that the model got better was further training. Adding more epochs onto the training continued to raise the validation rate all they way up until about 900 epochs. In the end the model was run and trained for 1000 epochs to make sure it would reach that point. The other method that was employed to try and achieve a higher validation accuracy was modifying the binning parameters used during pre-processing. Each group was split into more bins or less bins, as well as trying to find more even distributions or natural breaks to avoid borderline classification for some of the variables. If values fell near or on a bin border it could lead to higher errors due to inevitable variability with machine learning.

5 Results

The network reached a validation accuracy maximum at just over 97%. The network hovered around 90% validation accuracy for the majority of epochs beyond 500 epochs. Based on such a high validation accuracy we can be confident that our model has achieved our main goal. The model was able to give an accurate prediction of housing prices based on simple parameters that are easily attainable and implementable by the average home buyer.

6 Error Analysis

One of the main contributors to error is the existence of an abnormal market event within our data. This is due to the crash in the stock market, and the following recession in 2009. Market values experienced dramatic changes due to external factors that could not be explained by our model. Thus, our attempts to compensate for normal fluctuations in normal year to year swings in the market failed. Given more time and resources, we would want to collect more data similar to that which was used in this project from the past five years excluding 2020 (to avoid the effects of Covid-19). This would give a better assessment of our model's performance on predicting housing prices. With more work and data inputs it could be possible to provide the model data about trends in the market, or enough data that it recognizes them itself so then it could accurately predict even with abnormal markets.

7 References

- Keras documentation: https://keras.io/api/
- Our dataset was pulled from the following URL on Kaggle: https://www.kaggle.com/prevek18/ames-housing-dataset.
- Dataset info and paper by Dean DeCock: http://jse.amstat.org/v19n3/decock.pdf