INET 4710

Group 4

Final Group Project: Zillow's Home Value Prediction

Summary and problem definition for management

When in the market for a new home, there are many things a potential buyer has to take into consideration before deciding on one. A new house is an expensive asset that some people may have for their entire life. Due to this, it is important to ensure that the buyer is getting the best possible price from the features present in the house. The selling price of the house can drastically fluctuate from the listed price depending on factors such as features that were changed or changes to the surroundings like in the neighborhood.

With this being said, our model aims to solve this issue for people interested in buying a home through predicting the actual home price in the Twin Cities area. This will be done by using sentiment analysis in combination with a convolutional neural network to analyze TheMLSonline's metadata, the realtor's description, images of the house, and other factors such as appliances. Using all of this, the model should be able to produce a price that is accurate to the true home sale price.

Research design, measurement & statistical methods, traditional and machine learning methods

The methodology for modeling the housing prices is a regression model with a gradient boost to help ensure greater accuracy. The accuracy of the model was recorded for each data

“level” available to us. Starting with just the cleaned metadata, description sentiment analysis, and image data. Then combinations of these levels were recorded for as well. The metrics used for overall accuracy were adj-R squared, mean squared error, and mean absolute error. These were recorded for both the training and test sets to give a more complete picture of the model’s accuracy. The data was also scaled to accommodate the varying magnitudes between each column of the set.

Overview of programming work

Our model makes use of Jacob’s mls scraper which scrapes for information on properties in the Twin Cities area. The data gathered via this scraper is a huge mess of metadata which, by itself, is unworkable, and requires hefty amounts of cleaning. The data contains many missing/null values for specific columns, as well as some entire rows of houses to be null data. To combat this issue, we dropped all the null rows while filling in either 0 or “None” into the missing values respective to the columns data type of int or string. We also had other data preprocessing which got rid of unwanted data that are uninformative/repetitive, irrelevant, or are duplicates. There were also issues with the workability of many of the feature columns, combining their data into long strings separated by colons, making it useless without any work. We split these strings into new columns to account for each of the values present. For example, a column like “basement_details” would say if the basement was finished or not, had a sump pump, etc. Once we had all of the unique individual housing features, we could add a new column which indicated the absence or presence of these features, basically one-hot-encoding them. We also made the columns lowercase, just for added consistency. Afterwards, we used One Hot Encoding to create individual columns for each of the categorical features.

Descriptions of the houses were also processed with some Natural Language Processing to perform a sentiment analysis and see any relations between certain description words and our ending price. Since the scraped decisions were all condensed into single long strings with no spaces, we added spaces to make it easier to read while also making it easier for NLP algorithms to work on it. Lastly we had to scrape Google images for pictures at each of the addresses by running the address data through a scraper to save pictures taken from Google Image.

Review of results with recommendations for management

With the limited amount of images we could scrape, the best model is the model containing the metadata and the description data, giving an adj-R squared of 0.89. Here are the summary statistics for the best model:

```
adj r2 train =  0.9416017929318681
adj r2 test =  0.8864141799977091
RMSE train =  3225442287.7285376
RMSE test =  6198246634.499182
MAE train =  35751.78872020607
MAE test =  39135.1194224115
```

Having an adj-R squared of nearly 0.9 for the test set is extremely good considering the number of terms in the metadata and description dataset. Adj-R squared can fall off significantly as the number as the number of terms in a dataset increases, especially if they do not provide much in the way of improving model accuracy. What this means is that ~90% of the variance in the data can be explained by the model, giving us very accurate results for any future predictions that are made with our model.