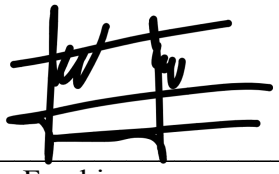
Quantifying the Relationship between Housing Prices and Permits in Austin, Texas

Completed for the Applied Statistical Modeling
Certificate Spring 2023

Vaishnavi Sathiyamoorthy
B.S Computational Biology
Department of Integrative Biology
College of Natural Sciences



Dr. Arya Farahi
Assistant Professor
Department of Statistics and Data Sciences

Abstract
The prices of homes in Austin, Texas have increased rapidly over the past decade due to a surge in both residential and commercial properties. This report aims to investigate whether residential and commercial permit data can be used to predict housing prices in Austin. To develop a prediction model, the Permits dataset for Austin and the Austin Multiple Listing Service (MLS) dataset were utilized. The permits were categorized as residential, commercial, important, and unimportant. Exploratory data analysis was conducted to examine the distribution of housing prices and permits between 2009 and 2018.

Using R, a new dataset was created that included the properties sold between 2009 and 2018 and the number of residential important, residential unimportant, commercial important, and commercial unimportant permits within a two-mile radius in 0.2-mile increments. This resulted in 40 features for each house. The dataset was cleaned and uploaded to Google Colaboratory. The `Keras` library in Python was used to develop the deep learning model. 80% of the data was used to train the model and 20% of the data was used to test the model. To train the Multilayer Perceptron (MLP) regression model, the logs of the selling prices were used. The best working model contained 3 hidden layers with rectified linear unit activation functions and the output layer with a linear activation function. The model's mean squared error was 0.09—a deviation of 9% from the actual value. This study demonstrates that residential and commercial permits issued in the surrounding area can determine the selling price of homes. Our results suggest that policies on permits issued in the Austin area could help control the rapid increase in property values.

Introduction
Housing prices have increased throughout the United States over the past decade. However, certain cities in the country have had the cost of homes have doubled. Austin, Texas has been one of the fastest growing metropolitan areas.[1] The median prices of homes sold each year has drastically increased in the past 15 years. The median selling price of homes in 2008 was approximately $185,000. In 2022, the number has surged to approximately $500,300.[2]

Austin has rapidly increased the amount of restaurants, trendy coffee shops, expensive bars, yoga studios, and luxury apartment complexes. There have been numerous demolitions of old homes and smaller apartments to build luxury apartment complexes. This has unfortunately displaced a huge number of low income families and has severely impacted the Black and Latinx community.[3]

One of the biggest reasons associated with growth in the city is because major technology companies, such as Apple, IBM, Tesla, Amazon, and AT&T, have moved into the metropolitan area. This has led to wealthier and younger individuals moving into the Austin area. The median household income has also increased from $55,744 in 2010 to $80,954 in 2020.[3]

Due to the increase in high paying jobs, there has been a huge influx of individuals moving into the Austin area. In fact, the population has increased by 160,000 in the past ten years. The population growth has led to a shortage of homes. Since the demand for homes is higher than the supply of homes, numerous homes have been sold for higher than the asking price.[3]

Individuals buying homes want to be in close proximity to their jobs. As expected, there would be an increase in the number of commercial and residential properties that are built near sectors of high paying jobs. These properties require permits when beginning construction. When analyzing the number of construction permits for single family homes in Texas, there has been a steady increase from 2010 to 2022.[4] Thus, there seems to be a trend in the population influx, increased demand for homes, and the number of permits issued.

Due to the increased costs in housing, this has led to a housing crisis for low income families in the Austin area. Latinx and Black communities have severely been impacted from gentrification and displacement. The housing crisis in Austin has led to numerous policy proposals to help those who are severely affected.[5] Analyzing the impact that residential and commercial permits have on the cost of housing in Austin could help drive housing policy changes.

The above literature shows the overall trend in increased housing costs and the number of permits in Austin, Texas. This justifies the need to study whether the number of permits issued near the area of the homes can predict the selling price of the home. The purpose of this study was to test the hypothesis that there is a relationship between residential and commercial permits and the housing market in Austin, Texas. It was predicted that the number of residential and commercial permits issued could determine the selling price of the home in Austin, Texas.

Data

Data Cleaning

The Austin Multiple Listing Service (MLS) dataset contained information on all the houses that were sold between 1996 and 2018. This dataset was acquired from the MLS database.[6] The permits dataset contained information on all the residential and commercial permits that were issued in the City of Austin between 1921 and 2021. This dataset was acquired from the GIS portal of the City of Austin.[7]

R was used to clean and join the datasets.[8] The packages that were used were `tidyverse`, `geosphere`, `lubridate`, `stringr` and `readr`.[9, 10, 11, 12, 13] The permits were categorized as residential and commercial. The only permits that were taken into consideration were those whose current status was “final”, “active”, or “closed”. These permits were also categorized as

important and unimportant. The important permits were building, mechanical, and electrical. The unimportant permits were driveway/sidewalk and plumbing.

A new dataframe was created that had the number of residential important, residential unimportant, commercial important, and commercial unimportant permits within a two mile radius in 0.2 mile increments and within 365 days before the listing of the house. A for loop was created so that each house from the MLS dataframe could be associated with commercial, residential, important, and unimportant features. During each loop in the for loop, a temporary distance column was created to determine the distance between the house and each permit using geodist. This data was converted to miles. A temporary days column was also created to determine the number of days between the permit date and the house listing date.

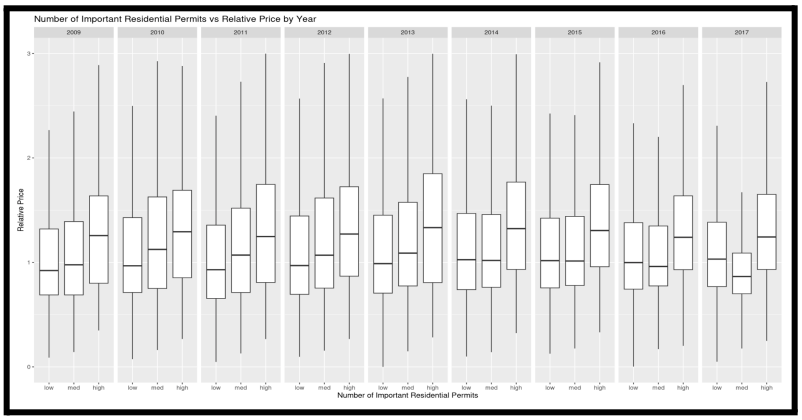
A temporary dataset was created that filtered the permits to residential, important, and within 365 days of when the house was listed and the permit was completed. This was done using tidyverse. This information was split into ten columns by looking at the distance in 0.2 mile increments up until a two mile radius. The above process was done for residential unimportant, commercial important, and commercial unimportant. This resulted in 40 features. Each row of this dataset that was created had the MLS number for each house.

The dataset that was created was joined with the MLS dataset's square feet, selling price, year built, and year sold. The two datasets were joined by the MLS number. This final dataset was modified by adding a column that included that average housing price in the previous year of that month. Another column was also added with the month of the year. The natural log of the selling price and the natural log of the average housing price in the previous year of that month was taken. This was the final dataset that was used for exploratory data analysis and to train the multilayer perceptron (MLP) regression model.

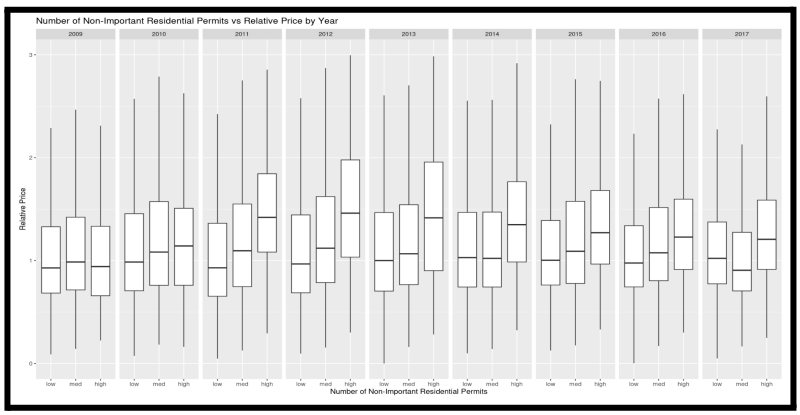
Exploratory Data Analysis

When analyzing the data initially, the amount of permits were arbitrarily categorized as low, medium, and high for residential important, residential unimportant, commercial important, and commercial unimportant. Each of the homes were given a relative price which was calculated by dividing the selling price by the average price of the homes from the previous year. When comparing each of the categories against the relative selling price, there seems to be a general trend where lower relative prices have a lower number of permits within a two mile radius and higher relative prices have a higher number of permits within a two mile radius. The high, medium, and low categories for the number of permits for each category was determined by looking at the distribution of the number of permits surrounding each house. These distributions were approximately, evenly split into 3. The boxplots in Figure 1 displays this visual relationship.

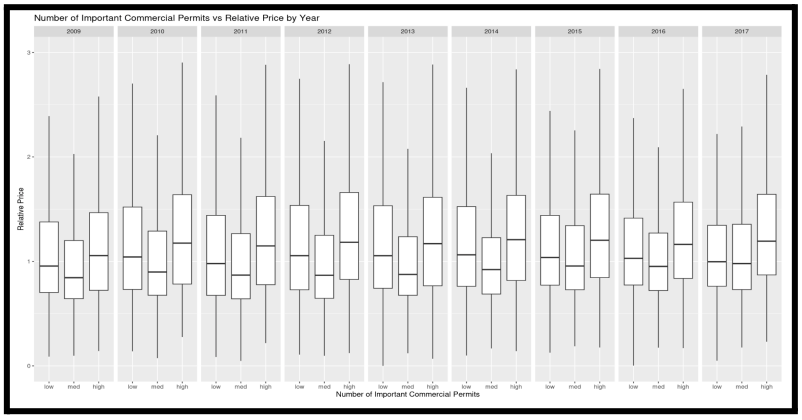
a)

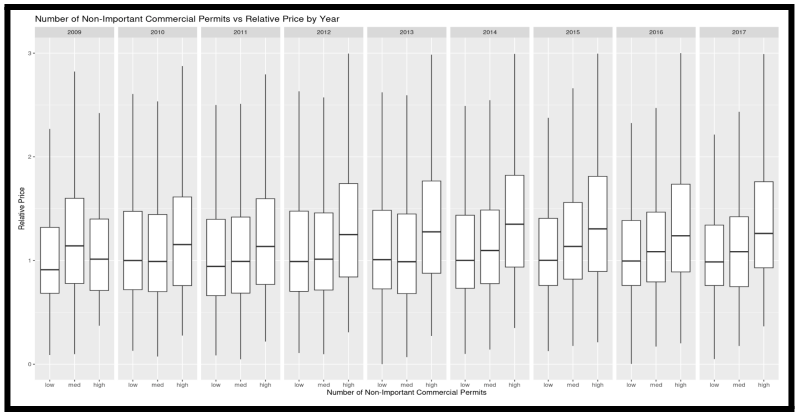

Number of Important Residential Permits vs Relative Price by Year
2009
2010
2011
2012
2013
2014
2015
2016
2017
Relative Price
0
1
2
3
low
med
high
Number of Important Residential Permits

b)


Number of Non-Important Residential Permits vs Relative Price by Year
2009
2010
2011
2012
2013
2014
2015
2016
2017
Relative Price
0
1
2
3
low
med
high
Number of Non-Important Residential Permits

c)

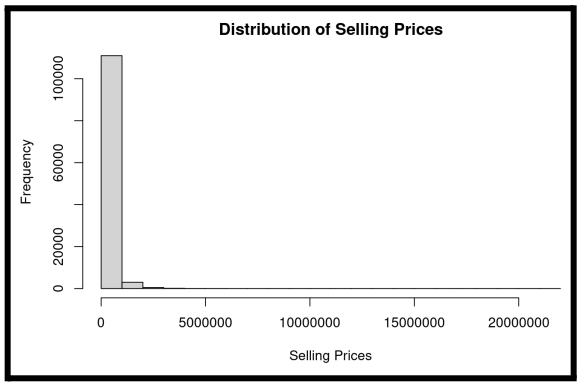

Number of Important Commercial Permits vs Relative Price by Year
2009
2010
2011
2012
2013
2014
2015
2016
2017
Relative Price
0
1
2
3
low
med
high
Number of Important Commercial Permits



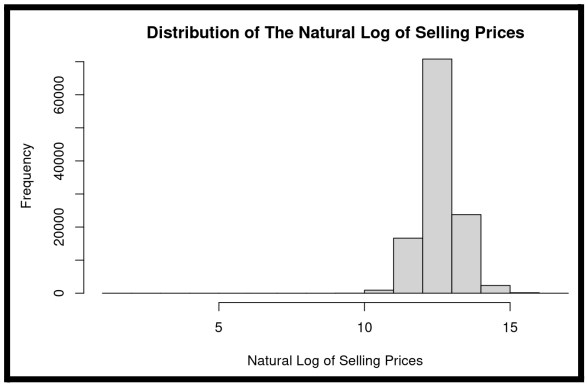
d)

Figure 1. The distribution of relative price of the homes for permit amounts of low, medium, and high are shown for each year between 2009 and 2018. **a)** This shows the residential important distributions. **b)** This shows the residential unimportant distributions. **c)** This shows the commercial important distributions. **d)** This shows the commercial unimportant distributions.

This distribution of the list prices was right skewed. To better train the model, the log of the prices was taken. This created a normal distribution of the list prices. The distributions are shown in Figure 2.



a)



b)

Figure 2. a) The distribution of the selling prices is shown in dollars. **b)** The distribution of the natural log of the selling prices is shown.

After the initial analysis of the data, the MLP regression model could be created.

Method

The final dataset contained information for each house in the MLS dataset. This included the MLS number, the square footage of the home, the selling date, the selling year, the selling month, the average selling price for homes for that month in the previous year, the number of residential important, residential unimportant, commercial important, and commercial unimportant permits for each 0.2 mile increment within a 2 mile radius, and the log of the selling price.

The regression model was created using Python.[14] `Numpy`, `pandas`, `keras`, `sklearn`, and `matplotlib` were imported.[15, 16, 17, 18, 19] `Pandas` was used to import the data frame. The data was split into the input variables (the square footage of the home, the selling date, the selling year, the selling month, the natural log of the average selling price for homes for that month in the previous year, and the number of residential important, residential unimportant, commercial important, and commercial unimportant permits for each 0.2 mile increment within a 2 mile radius) and the output variable (selling price). Using `sklearn`, the data was randomly split into training and testing datasets. 80% of the data was used to train the model, while 20% of the data was used to test the model.

Using `Keras`, the MLP regression model was built. This is a neural network algorithm that tackles the problem of training non-linear data. When the algorithm initially starts, arbitrary weights are given to each of the neurons. There is an input layer, one or more hidden layers, and an output layer. The calculations done in each layer are fed into the next layer. To ensure learning, backpropagation occurs. This is where in each iteration, the weights of the neurons are adjusted to reduce the loss function.[20] The loss function that was used for this model was mean squared error (MSE) and the equation is shown in Figure 3.

$$MSE = \frac{1}{n}\Sigma\left(y - \widehat{y}\right)^2$$

Figure 3. The mean squared error equation is shown. n refers to the number of samples. y refers to the actual value. ŷ refers to the value predicted by the model.

The weights mentioned previously were adjusted using the Adam optimization algorithm. This algorithm improves the performance of models where linearity is not assumed and takes the mean of the "magnitudes of the gradients of the weights."[21] Throughout the model, all layers except for the output layer had a rectified linear unit activation function. An activation function determines whether the neuron should be activated. A rectified linear unit (ReLU) function is a piecewise linear function where the output is either zero or positive. Studies show that this activation function leads to better training and has better performance.[22]

A sequential model was created. The input layer consisted of 40 neurons, a rectified linear unit activation function, 10 input dimensions, and an input shape of 45. The first hidden layer consisted of 30 neurons and a ReLU activation function. The second hidden layer consisted of 20 neurons and a ReLU function. The third hidden layer consisted of 10 neurons and a ReLU function. The output layer consisted of 1 output neuron and a linear activation function. Figure 4 shows the visual representation of the MLP regression model. When compiling the function, the

mean squared error was used to calculate the loss and the Adam optimizer was used. There were 150 iterations of the algorithm run to train the model. This model was fit to the training and testing datasets.

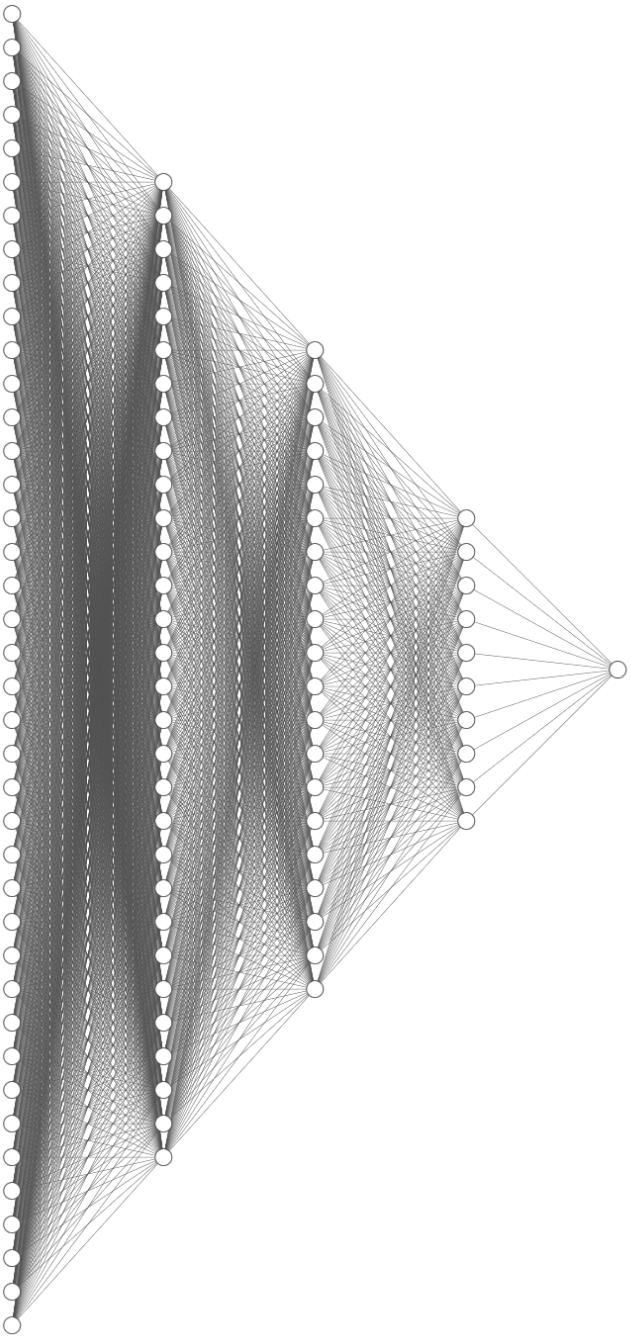


Figure 4. Visual representation of MLP regression model with one input layer, three hidden layers, and one output layer.[23]

The loss for the training and testing dataset were visualized using matplotlib. The mean absolute error, mean squared error, root mean squared error, and the variance score were calculated using `sklearn`. The testing data was plotted to show the accuracy of the model by plotting the actual natural log values for the selling price and the predicted natural log values of the selling price.

In order to determine if the number of residential and commercial permits were good predictors for the natural log of the selling price, a comparison model was created. This only included the input values of the square footage of the home, the selling date, the selling year, the selling month, and the natural log of the average selling price for homes for that month in the previous year. The output value was the natural log of the selling price of the home.

Using `Keras`, the MLP regression model was built. A sequential model was created. The input layer consisted of 40 neurons, a rectified linear unit activation function, 10 input dimensions, and an input shape of 4. The rest of the model was consistent with the original model. The same plots and calculations done for the original model were done for the comparison model.

Results

The MLP regression model that was trained with the permits data had a mean squared error of 0.09 for the testing data when taking the natural log of the selling price. This means that there was an average of a ~9% deviation from the actual price value and the predicted value. This value determines how well the model fits the data and how much the prediction deviates from the actual value. Since this value was low, it can be concluded that the model fit the data. The R^2 was 0.77, which meant that 77% of the variance in the the natural log of the selling price could be explained by the square footage of the home, the selling date, the selling year, the selling month, the natural log of the average selling price for homes for that month in the previous year, and the number of residential important, residential unimportant, commercial important, and commercial unimportant permits for each 0.2 mile increment within a 2 mile radius. Figure 5 shows that the mean squared error (loss) decreased for each iteration of the model. Figure 6 shows that predicted values are dispersed above and below the regression line.

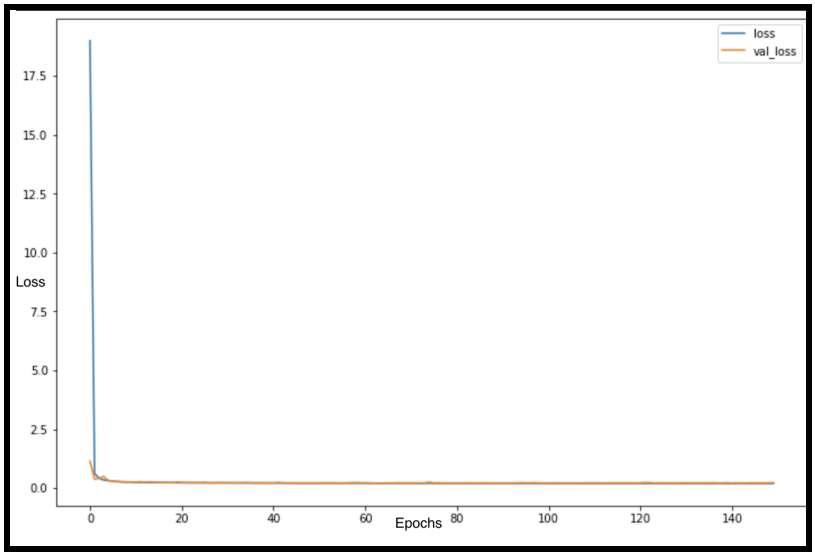


Figure 5. This graph shows the mean squared error (loss) for each iteration of the training and testing dataset with the permits data. The blue line shows the loss for the training dataset and the orange line represents the loss for the testing dataset.

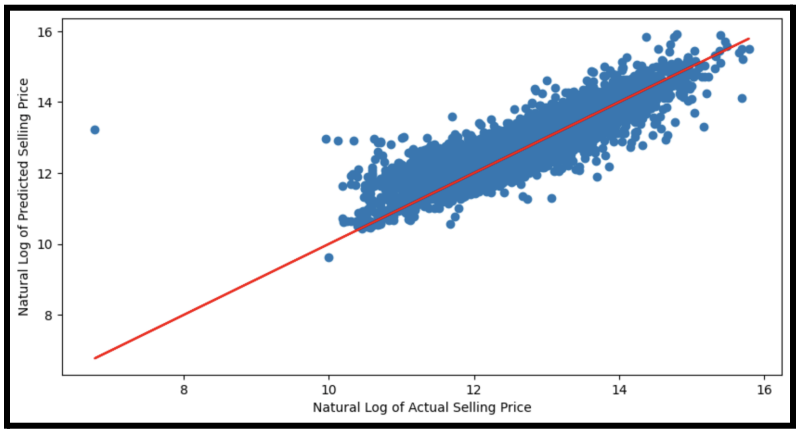


Figure 6. This scatter plot shows the natural log of the actual selling price of the homes on the x axis and the natural log of the predicted selling price of the homes on the y axis.

The MLP model that was trained without the permits data had a mean squared error of 0.17: 17% deviation from the actual value. This model did not fit the data as well as the model with the permits data. The R^2 was 0.57, which meant that 57% of the variance in the natural log of the selling price could be explained by the square footage of the home, the selling date, the selling year, the selling month, and the natural log of the average selling price for homes for that month in the previous year. Figure 7 shows that the mean squared error (loss) decreased for each iteration of the model. Figure 8 shows that predicted values are dispersed above and below the regression line. The scatterplot shows that the model tended to underestimate the value of the homes.

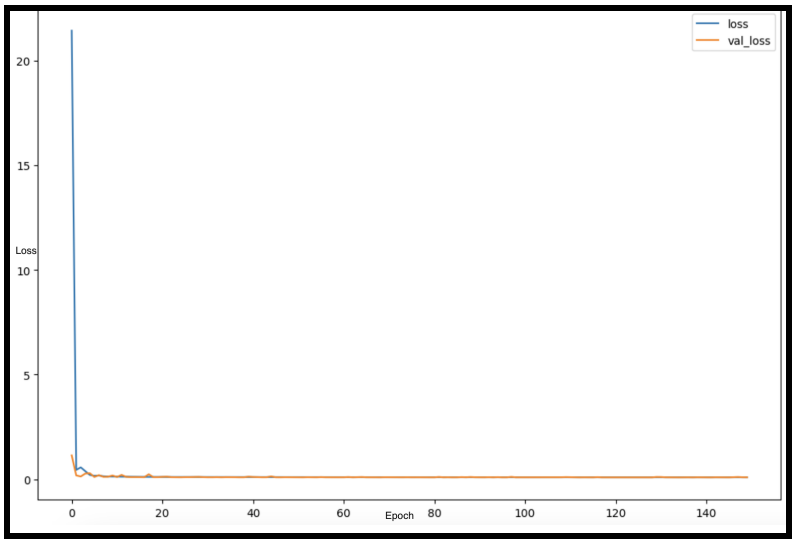


Figure 7. This graph shows the mean squared error (loss) for each iteration of the training and testing datasets without the permits data. The blue line shows the loss for the training dataset and the orange line represents the loss for the testing dataset.

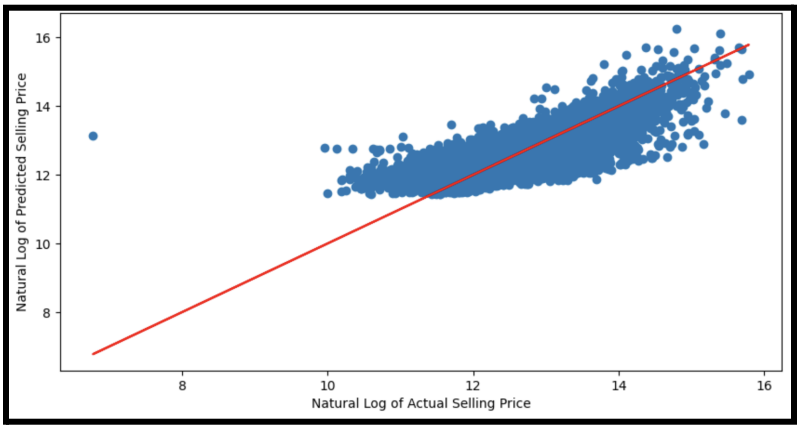


Figure 8. This scatter plot shows the natural log of the actual selling price of the homes on the x axis and the natural log of the predicted selling price of the homes on the y axis.

Discussion

The purpose of this study was to test the hypothesis that there is a relationship between residential and commercial permits and the housing market in Austin, Texas using MLP Regression. The model developed with the permits data had a mean squared error of 0.09 for the testing dataset. This was significantly less when compared to the model developed without the permits dataset: 0.17. The variance score was also higher for the model with the permits data. When looking at Figure 6 and Figure 8 from the previous section, the model with the permits data has data points evenly scattered above and below the regression line. The model without the permits data tended to underestimate the housing value.

When comparing these results with the exploratory data analysis, it follows that initial prediction that the number of permits correlates with the selling price of the home. These results also support the literature mentioned earlier in this report. The increase in commercial and residential properties in Austin have increased the property values of homes in the same neighborhood. Thus, it can be concluded that permits data could be used to predict the values of neighborhood homes in Austin.

To display the effect of permits on the selling prices in Austin, two separate datasets were created with only 2017 Austin housing data. One of the datasets had the original number of permits, while the other had 50% of the number of permits in each category. When comparing the predicted values, the average natural log of the predicted selling price with the original number of permits was 12.619514 and the average natural log of the predicted selling price with the reduced number of permits was 12.432438. When converting these values to the selling price, that is a change from $302,402.60 to $250,806.72. Thus, permits imply an increas of over $50,000, on average. The distribution of the predicted natural logs of the selling prices can be seen in Figure 9.

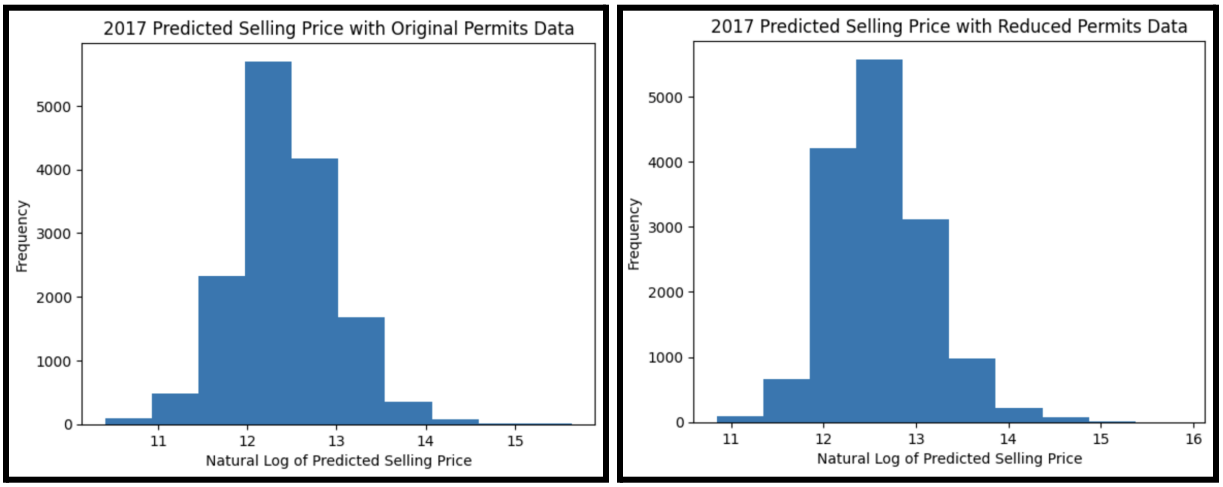


Figure 9. The first plot shows the distribution of the predicted natural logs of the selling price with the original permits data. The second plot shows the distribution of the predicted natural logs of the selling price with the reduced permits data.

This study has significant implications in terms of housing policy in Austin. Residential and commercial permit regulations could be modified in order to slow the growth in property values. In addition, this model could be used to predict the selling price of homes in the city, which would be useful for those selling homes. Finally, the impact of building new residential and commercial properties can be seen through this model.

While this study did have promising results, there were certain limitations. While the value of homes in Austin has been increasing since 2009, there has been a much more significant increase in the past five years. The model includes that from 2009, which could decrease the accuracy of the models. In addition, numerous data points were removed due to NA values in the rows. Computations could have been done to determine those values in order to increase the amount of data used to train the model. Finally, it cannot be determined which features used to train the model most significantly influence the outputs.

Further research needs to be done in Austin housing prices and factors that influence them. In this study, it is not determined which permit categories have the most influence in the selling price in Austin. More research in this area could determine this aspect. In addition, this study focuses on housing prices within the city limits of Austin. There has been a significant increase in the selling prices of homes in the surrounding areas, such as Round Rock, Pflugerville, Cedar Park, Lakeway, and Bee Cave. Expanding this study to these areas could be crucial in predicting the selling prices.

Conclusion

The price of homes in Austin, Texas has increased rapidly over the past decade. The factors that influence home values are crucial to predicting the selling price of homes. This study showed that residential and commercial permits are key in predicting property values. This has implications for housing policy in the city of Austin. Further research needs to be done on the permit categories that influence the selling price the most.

References

1. Sharf, S. (2018, February 28). *Full list: America's fastest-growing cities 2018*. Forbes. Retrieved March 30, 2023, from https://www.forbes.com/sites/samanthasharf/2018/02/28/full-list-americas-fastest-growing-cities-2018/?sh=26fa5d0a7feb
2. *Housing Activity for Austin-Round Rock*. Texas Real Estate Research Center. (n.d.). Retrieved March 30, 2023, from https://www.recenter.tamu.edu/data/housing-activity/#!/activity/MSA/Austin-Round_Rock
3. Sandoval, E. (2021, November 27). *How Austin became one of the least affordable cities in America*. The New York Times. Retrieved March 30, 2023, from https://www.nytimes.com/2021/11/27/us/austin-texas-unaffordable-city.html
4. Roberson, J., Yan, W., & Shaunfield, J. (2023, March 20). *Texas Housing Insight*. Texas Real Estate Research Center. Retrieved March 30, 2023, from https://www.recenter.tamu.edu/articles/technical-report/Texas-Housing-Insight
5. *Austin Housing Analysis*. ArcGIS StoryMaps. (2021, August 15). Retrieved March 30, 2023, from https://storymaps.arcgis.com/collections/b1d4ecbe56d1427691045ca6b975db79?item=3
6. MLS Multiple Listing Service Listings. (n.d.). Retrieved from https://www.mls.com/.
7. Issued Construction Permits. (n.d.). Retrieved from https://data.austintexas.gov/dataset/Issued-Construction-Permits/3syk-w9eu.
8. R Core Team (2021). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
9. Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M,Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). “Welcome to the tidyverse.” _Journal of Open Source Software_, *4*(43), 1686. doi:10.21105/joss.01686.
10. Robert J. Hijmans (2022). *Geosphere: Spherical Trigonometry*. R package version 1.5-18. https://CRAN.R-project.org/package=geosphere.
11. Garrett Grolemund, Hadley Wickham (2011). *Dates and Times Made Easy with lubridate. Journal of Statistical Software*, 40(3), 1-25. URL https://www.jstatsoft.org/v40/i03/.
12. Hadley Wickham (2022). *stringr: Simple, Consistent Wrappers for Common String Operations.* R package version 1.5.0. https://CRAN.R-project.org/package=stringr.
13. Hadley Wickham, Jim Hester and Jennifer Bryan (2022). *readr: Read Rectangular Text Data.* R package version 2.1.3. https://CRAN.R-project.org/package=readr.
14. Van Rossum, G., & Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

15. Harris, C.R., Millman, K.J., van der Walt, S.J., Virtanen, P., Gommers, R., Oliphant, T.E., ... & Vandewalle, G. (2020). *Array programming with NumPy*. Nature, 585(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2.
16. McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).
17. Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras.
18. Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.
19. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90–95.
20. Taud, H., Mas, J. (2018). Multilayer Perceptron (MLP). In: Camacho Olmedo, M., Paegelow, M., Mas, JF., Escobar, F. (eds) Geomatic Approaches for Modeling Land Change Scenarios. Lecture Notes in Geoinformation and Cartography. Springer, Cham. https://doi.org/10.1007/978-3-319-60801-3_27.
21. Brownlee, J. (2021, January 12). *Gentle Introduction To The Adam Optimization Algorithm For Deep Learning*. Machine Learning Mastery. Retrieved April 5, 2023, from https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/
22. Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU).
23. NN-SVG. (n.d.). Retrieved from http://alexlenail.me/NN-SVG/index.html

Appendix

Reshaping Code (R)

```
library(dplyr)
library(tidyverse)
library(geosphere)
library(lubridate)
library(stringr)

mls <- read_csv("~/SDS379R/austin_mls_all_1996_2018.csv")
permits <- read_csv("~/SDS379R/Issued_Construction_Permits.csv")
permits_selected <- permits %>% select(c(`Permit Type`, `Permit Num`, `Permit Class Mapped`, `Permit Class`, `TCAD ID`, `Applied Date`, `Issued Date`, `Calendar Year Issued`, `Fiscal Year Issued`, `Status Current`, `Status Date`, `Expires Date`, `Completed Date`, `Original Address 1`, `Original City`, `Original State`, `Original Zip`, `Project ID`, Latitude, Longitude))

permits_residential <- permits_selected %>% filter(`Permit Class Mapped` == "Residential")
permits_commercial <- permits_selected %>% filter(`Permit Class Mapped` == "Commercial")

permits_residential <- permits_residential %>% filter(`Status Current` == "Final" | `Status Current` == "Active" | `Status Current` == "Closed")
permits_commercial <- permits_commercial %>% filter(`Status Current` == "Final" | `Status Current` == "Active" | `Status Current` == "Closed")

permits_residential$Importance <- NA
permits_residential$Importance[permits_residential$`Permit Type` == "BP" |
permits_residential$`Permit Type` == "MP" |permits_residential$`Permit Type` == "EP"] <- "important"
permits_residential$Importance[permits_residential$`Permit Type` == "DS" |
permits_residential$`Permit Type` == "PP"] <- "nonimportant"

permits_commercial$Importance <- NA
permits_commercial$Importance[permits_commercial$`Permit Type` == "BP" |
permits_commercial$`Permit Type` == "MP" |permits_commercial$`Permit Type` == "EP"] <- "important"
permits_commercial$Importance[permits_commercial$`Permit Type` == "DS" |
permits_commercial$`Permit Type` == "PP"] <- "nonimportant"

permits_commercial <- permits_commercial %>% na.omit
permits_residential <- permits_residential %>% na.omit
```

```
mls <- mls %>% select(c(3,4,5,13,14,16,22,23,24,25))
mls$`List Date` <- as.Date(mls$`List Date`,format="%m/%d/%Y")

mls <- rename(mls, "Latitude_mls" = "Latitude")
mls <- rename(mls, "Longitude_mls" = "Longitude")
permits_residential <- rename(permits_residential, "Latitude_perm" = "Latitude")
permits_residential <- rename(permits_residential, "Longitude_perm" = "Longitude")

permits_commercial <- rename(permits_commercial, "Latitude_perm" = "Latitude")
permits_commercial <- rename(permits_commercial, "Longitude_perm" = "Longitude")

permits_selected <- rbind(permits_residential, permits_commercial)
mls <- mls %>% filter(Year >= 2008)
permits_selected <- permits_selected %>% filter(`Calendar Year Issued` >= 2008)

new_data_frame <- data.frame(matrix(ncol = 41, nrow = 0))

colnames(new_data_frame) <- c("MLS #", "Residential_Important_0-0.2",
"Residential_Important_0.21-0.4", "Residential_Important_0.41-0.6",
"Residential_Important_0.61-0.8", "Residential_Important_0.81-1",
"Residential_Important_1.01-1.2", "Residential_Important_1.21-1.4",
"Residential_Important_1.41-1.6", "Residential_Important_1.61-1.8",
"Residential_Important_1.81-2.0", "Residential_Non-Important_0-0.2",
"Residential_Non-Important_0.21-0.4", "Residential_Non-Important_0.41-0.6",
"Residential_Non-Important_0.61-0.8", "Residential_Non-Important_0.81-1",
"Residential_Non-Important_1.01-1.2", "Residential_Non-Important_1.21-1.4",
"Residential_Non-Important_1.41-1.6", "Residential_Non-Important_1.61-1.8",
"Residential_Non-Important_1.81-2.0", "Commercial_Important_0-0.2",
"Commercial_Important_0.21-0.4", "Commercial_Important_0.41-0.6",
"Commercial_Important_0.61-0.8", "Commercial_Important_0.81-1",
"Commercial_Important_1.01-1.2", "Commercial_Important_1.21-1.4",
"Commercial_Important_1.41-1.6", "Commercial_Important_1.61-1.8",
"Commercial_Important_1.81-2.0", "Commercial_Non-Important_0-0.2",
"Commercial_Non-Important_0.21-0.4", "Commercial_Non-Important_0.41-0.6",
"Commercial_Non-Important_0.61-0.8", "Commercial_Non-Important_0.81-1",
"Commercial_Non-Important_1.01-1.2", "Commercial_Non-Important_1.21-1.4",
"Commercial_Non-Important_1.41-1.6", "Commercial_Non-Important_1.61-1.8",
"Commercial_Non-Important_1.81-2.0")
for (mls_data in 1:125160) {
```

```
    lat_mls <- mls[mls_data,5]
    long_mls <- mls[mls_data,4]
    date_mls <- mls[mls_data,7]
    temp <- permits_selected
    temp$mls_lat <- lat_mls
    temp$mls_long <- long_mls
    temp$mls_date <- date_mls
    temp$diff_date <- as.numeric(temp$mls_date$`List Date` - temp$`Completed Date`)
    temp$dist <- geodist::geodist_vec(x1 = unlist(temp$mls_long$Longitude_mls), y1 =
unlist(temp$mls_lat$Latitude_mls), x2 = unlist(temp$Longitude_perm), y2 =
unlist(temp$Latitude_perm), paired = TRUE, measure = "haversine")
    temp$dist <- temp$dist / 1609.34
    new_data_frame[mls_data,1] <- mls[mls_data,1]
    temp_1 <- temp %>% filter(`Permit Class Mapped` == "Residential") %>% filter(Importance
== "important") %>% filter(diff_date >= 0 & diff_date <= 365)
    new_data_frame[mls_data,2] <- temp_1 %>% filter(dist <= 0.2) %>% nrow
    new_data_frame[mls_data,3] <- temp_1 %>% filter(dist > 0.2 & dist <= 0.4) %>% nrow
    new_data_frame[mls_data,4] <- temp_1 %>% filter(dist > 0.4 & dist <= 0.6) %>% nrow
    new_data_frame[mls_data,5] <- temp_1 %>% filter(dist > 0.6 & dist <= 0.8) %>% nrow
    new_data_frame[mls_data,6] <- temp_1 %>% filter(dist > 0.8 & dist <= 1) %>% nrow
    new_data_frame[mls_data,7] <- temp_1 %>% filter(dist > 1 & dist <= 1.2) %>% nrow
    new_data_frame[mls_data,8] <- temp_1 %>% filter(dist > 1.2 & dist <= 1.4) %>% nrow
    new_data_frame[mls_data,9] <- temp_1 %>% filter(dist > 1.4 & dist <= 1.6) %>% nrow
    new_data_frame[mls_data,10] <- temp_1 %>% filter(dist > 1.6 & dist <= 1.8) %>% nrow
    new_data_frame[mls_data,11] <- temp_1 %>% filter(dist > 1.8 & dist <= 2) %>% nrow

    temp_2 <- temp %>% filter(`Permit Class Mapped` == "Residential") %>% filter(Importance
== "nonimportant") %>% filter(diff_date >= 0 & diff_date <= 365)
    new_data_frame[mls_data,12] <- temp_2 %>% filter(dist <= 0.2) %>% nrow
    new_data_frame[mls_data,13] <- temp_2 %>% filter(dist > 0.2 & dist <= 0.4) %>% nrow
    new_data_frame[mls_data,14] <- temp_2 %>% filter(dist > 0.4 & dist <= 0.6) %>% nrow
    new_data_frame[mls_data,15] <- temp_2 %>% filter(dist > 0.6 & dist <= 0.8) %>% nrow
    new_data_frame[mls_data,16] <- temp_2 %>% filter(dist > 0.8 & dist <= 1) %>% nrow
    new_data_frame[mls_data,17] <- temp_2 %>% filter(dist > 1 & dist <= 1.2) %>% nrow
    new_data_frame[mls_data,18] <- temp_2 %>% filter(dist > 1.2 & dist <= 1.4) %>% nrow
    new_data_frame[mls_data,19] <- temp_2 %>% filter(dist > 1.4 & dist <= 1.6) %>% nrow
    new_data_frame[mls_data,20] <- temp_2 %>% filter(dist > 1.6 & dist <= 1.8) %>% nrow
    new_data_frame[mls_data,21] <- temp_2 %>% filter(dist > 1.8 & dist <= 2) %>% nrow
```

```
   temp_3 <- temp %>% filter(`Permit Class Mapped` == "Commercial") %>% filter(Importance
== "important") %>% filter(diff_date >= 0 & diff_date <= 365)
   new_data_frame[mls_data,22] <- temp_3 %>% filter(dist <= 0.2) %>% nrow
   new_data_frame[mls_data,23] <- temp_3 %>% filter(dist > 0.2 & dist <= 0.4) %>% nrow
   new_data_frame[mls_data,24] <- temp_3 %>% filter(dist > 0.4 & dist <= 0.6) %>% nrow
   new_data_frame[mls_data,25] <- temp_3 %>% filter(dist > 0.6 & dist <= 0.8) %>% nrow
   new_data_frame[mls_data,26] <- temp_3 %>% filter(dist > 0.8 & dist <= 1) %>% nrow
   new_data_frame[mls_data,27] <- temp_3 %>% filter(dist > 1 & dist <= 1.2) %>% nrow
   new_data_frame[mls_data,28] <- temp_3 %>% filter(dist > 1.2 & dist <= 1.4) %>% nrow
   new_data_frame[mls_data,29] <- temp_3 %>% filter(dist > 1.4 & dist <= 1.6) %>% nrow
   new_data_frame[mls_data,30] <- temp_3 %>% filter(dist > 1.6 & dist <= 1.8) %>% nrow
   new_data_frame[mls_data,31] <- temp_3 %>% filter(dist > 1.8 & dist <= 2) %>% nrow

   temp_4 <- temp %>% filter(`Permit Class Mapped` == "Commercial") %>% filter(Importance
== "nonimportant") %>% filter(diff_date >= 0 & diff_date <= 365)
   new_data_frame[mls_data,32] <- temp_4 %>% filter(dist <= 0.2) %>% nrow
   new_data_frame[mls_data,33] <- temp_4 %>% filter(dist > 0.2 & dist <= 0.4) %>% nrow
   new_data_frame[mls_data,34] <- temp_4 %>% filter(dist > 0.4 & dist <= 0.6) %>% nrow
   new_data_frame[mls_data,35] <- temp_4 %>% filter(dist > 0.6 & dist <= 0.8) %>% nrow
   new_data_frame[mls_data,36] <- temp_4 %>% filter(dist > 0.8 & dist <= 1) %>% nrow
   new_data_frame[mls_data,37] <- temp_4 %>% filter(dist > 1 & dist <= 1.2) %>% nrow
   new_data_frame[mls_data,38] <- temp_4 %>% filter(dist > 1.2 & dist <= 1.4) %>% nrow
   new_data_frame[mls_data,39] <- temp_4 %>% filter(dist > 1.4 & dist <= 1.6) %>% nrow
   new_data_frame[mls_data,40] <- temp_4 %>% filter(dist > 1.6 & dist <= 1.8) %>% nrow
   new_data_frame[mls_data,41] <- temp_4 %>% filter(dist > 1.8 & dist <= 2) %>% nrow
}
new_data_frame <- inner_join(mls, new_data_frame, by = "MLS #")
write.csv(new_data_frame, "~/SDS379R/Reshaped_Datasets/reshape.csv")

df <- read_csv("~/SDS379R/final_dataframe.csv")
df$avg_price_month_prev_year <- NA
for (y in 2009:2018){
  for (m in 1:12){
   df$avg_price_month_prev_year[df$Year == y & df$month == m] <- df %>% filter(Year == (y
- 1)) %>% filter(month == m) %>%
     summarise(median(`Sold/Lease Price`)) %>% as.numeric
  }
}
df <- df %>% filter(Year != 2008)
sapply(df, class)
```

```
df <- df %>% select(-1)
mls <- read_csv("~/SDS379R/austin_mls_all_1996_2018.csv")
mls <- mls %>% select(`MLS #`, `Sqft Total`)
df <- inner_join(mls, df)
write_csv(df, "~/SDS379R/housing_permits.csv")
```

Box Plots (R)

```
Final_data_set <- read_csv("~/SDS379R/Reshaped_Datasets/reshape.csv")
final_data_set$Year <- as.numeric(final_data_set$Year)
final_data_set$`Sold/Lease Price` <- as.numeric(parse_number(final_data_set$`Sold/Lease
Price`))
final_data_set %>% group_by(Year) %>% summarize(median(`Sold/Lease Price`))
final_data_set$prev_med_property_value <- NA
final_data_set$prev_med_property_value[final_data_set$Year == 2009] <- 225000
final_data_set$prev_med_property_value[final_data_set$Year == 2010] <- 217000
final_data_set$prev_med_property_value[final_data_set$Year == 2011] <- 229000
final_data_set$prev_med_property_value[final_data_set$Year == 2012] <- 232000
final_data_set$prev_med_property_value[final_data_set$Year == 2013] <- 247900
final_data_set$prev_med_property_value[final_data_set$Year == 2014] <- 269000
final_data_set$prev_med_property_value[final_data_set$Year == 2015] <- 295000
final_data_set$prev_med_property_value[final_data_set$Year == 2016] <- 322500
final_data_set$prev_med_property_value[final_data_set$Year == 2017] <- 340990
final_data_set$price_obs_i_t <- final_data_set$`Sold/Lease Price` /
final_data_set$prev_med_property_value
final_data_set$total_res_imp <- final_data_set %>% select(11:20) %>% rowSums()
final_data_set$total_res_nonimp <- final_data_set %>% select(21:30) %>% rowSums()
final_data_set$total_com_imp <- final_data_set %>% select(31:40) %>% rowSums()
final_data_set$total_com_nonimp <- final_data_set %>% select(41:50) %>% rowSums()
# bins: 0-1000, 1001-2000, 2001-max
hist(final_data_set$total_res_imp)
# bins: 0-500, 501-1000, 1001-max
hist(final_data_set$total_res_nonimp)
# bins: 0-500, 501-1000, 1001-max
hist(final_data_set$total_com_imp, breaks = 20)
# bins: 0-200, 201-400, 401-max
hist(final_data_set$total_com_nonimp)
final_data_set$res_imp_cat <- NA
final_data_set$res_imp_cat[final_data_set$total_res_imp <= 1000] <- "low"
final_data_set$res_imp_cat[final_data_set$total_res_imp > 1000 & final_data_set$total_res_imp
<= 2000] <- "med"
```

```
final_data_set$res_imp_cat[final_data_set$total_res_imp > 2000] <- "high"
final_data_set$res_nonimp_cat <- NA
final_data_set$res_nonimp_cat[final_data_set$total_res_nonimp <= 500] <- "low"
final_data_set$res_nonimp_cat[final_data_set$total_res_nonimp > 500 &
final_data_set$total_res_nonimp <= 1000] <- "med"
final_data_set$res_nonimp_cat[final_data_set$total_res_nonimp > 1000] <- "high"
final_data_set$com_imp_cat <- NA
final_data_set$com_imp_cat[final_data_set$total_com_imp <= 500] <- "low"
final_data_set$com_imp_cat[final_data_set$total_com_imp > 500 &
final_data_set$total_com_imp <= 1000] <- "med"
final_data_set$com_imp_cat[final_data_set$total_com_imp > 1000] <- "high"
final_data_set$com_nonimp_cat <- NA
final_data_set$com_nonimp_cat[final_data_set$total_com_nonimp <= 200] <- "low"
final_data_set$com_nonimp_cat[final_data_set$total_com_nonimp > 200 &
final_data_set$total_com_nonimp <= 400] <- "med"
final_data_set$com_nonimp_cat[final_data_set$total_com_nonimp > 400] <- "high"

final_data_set %>% filter(price_obs_i_t <= 3) %>% ggplot(aes(x=factor(res_imp_cat,
levels=c('low', 'med', 'high')), y=price_obs_i_t)) + geom_boxplot(outlier.shape = NA) +
xlab("Number of Important Residential Permits") + ylab("Relative Price") + facet_grid(~Year) +
ggtitle("Number of Important Residential Permits vs Relative Price by Year")

final_data_set %>% filter(price_obs_i_t <= 3) %>% ggplot(aes(x=factor(res_nonimp_cat,
levels=c('low', 'med', 'high')), y=price_obs_i_t)) + geom_boxplot(outlier.shape = NA) +
xlab("Number of Non-Important Residential Permits") + ylab("Relative Price") +
facet_grid(~Year) + ggtitle("Number of Non-Important Residential Permits vs Relative Price by
Year")

final_data_set %>% filter(price_obs_i_t <= 3) %>% ggplot(aes(x=factor(com_imp_cat,
levels=c('low', 'med', 'high')), y=price_obs_i_t)) + geom_boxplot(outlier.shape = NA) +
xlab("Number of Important Commercial Permits") + ylab("Relative Price") + facet_grid(~Year)
+ ggtitle("Number of Important Commercial Permits vs Relative Price by Year")

final_data_set %>% filter(price_obs_i_t <= 3) %>% ggplot(aes(x=factor(com_nonimp_cat,
levels=c('low', 'med', 'high')), y=price_obs_i_t)) + geom_boxplot(outlier.shape = NA) +
xlab("Number of Non-Important Commercial Permits") + ylab("Relative Price") +
facet_grid(~Year) + ggtitle("Number of Non-Important Commercial Permits vs Relative Price by
Year")
```

Natural Log vs Selling Price Histograms (R)

```
hist(df$natural_log, main = "Distribution of The Natural Log of Selling Prices", xlab = "Natural Log of Selling Prices")
hist(df$`Sold/Lease Price`, main = "Distribution of Selling Prices", xlab = "Selling Prices")
```

MLP Models and Results (Python)

```
from google.colab import drive
drive.mount('/content/drive')
!pip install scikeras
import numpy as np
import pandas as pd
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
import matplotlib.pyplot as plt
%cd /content/drive/'My Drive'/TensorFlow/
#define function to swap columns. Code from https://www.statology.org/swap-columns-pandas/
def swap_columns(df, col1, col2):
  col_list = list(df.columns)
  x, y = col_list.index(col1), col_list.index(col2)
  col_list[y], col_list[x] = col_list[x], col_list[y]
  df = df[col_list]
  return df
dataframe = pd.read_csv("housing_permits.csv", delimiter=",")
dataframe = swap_columns(dataframe, "avg_price_month_prev_year", "Sold/Lease Price")
dataframe["ln_prev_year"] = np.log(dataframe["avg_price_month_prev_year"])
dataframe = swap_columns(dataframe, "ln_prev_year", "avg_price_month_prev_year")
dataframe = swap_columns(dataframe, "Sqft Total", "List Date")
dataframe = swap_columns(dataframe, "Sqft Total", "natural_log")
dataframe = dataframe.dropna()
# split into input (X) and output (Y) variables
X = dataframe[dataframe.columns[8:53]]
Y = dataframe[dataframe.columns[7]]
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=101)
model = Sequential()
model.add(Dense(40, input_dim=10, activation='relu', input_shape = (45,)))
model.add(Dense(30, activation='relu'))
model.add(Dense(20, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation = 'linear'))
```

```
model.compile(loss='mse', optimizer='Adam')
model.fit(X_train, Y_train, validation_data=(X_test, Y_test), epochs=150, batch_size = 32)
model.summary()
loss_df = pd.DataFrame(model.history.history)
loss_df
X_Test_2017 = dataframe[dataframe.columns[7:53]]
X_Test_2017 = X_Test_2017[X_Test_2017['Year'] == 2017]
Y_Test_2017 = X_Test_2017[X_Test_2017.columns[0]]
X_Test_2017 = X_Test_2017[X_Test_2017.columns[1:46]]

X_Test_2017_mod = X_Test_2017.copy()
X_Test_2017_mod[X_Test_2017_mod.columns[3:43]] =
X_Test_2017_mod[X_Test_2017_mod.columns[3:43]] * 0.5
# average log value = 12.619514. Average home price = 302402.6
y_pred_2017 = model.predict(X_Test_2017)
np.mean(y_pred_2017)
np.exp(np.mean(y_pred_2017))
# average log value = 12.432438. Average home price = 250806.72
y_pred_2017_mod = model.predict(X_Test_2017_mod)
np.mean(y_pred_2017_mod)
np.exp(np.mean(y_pred_2017_mod))

plt.hist(y_pred_2017)
plt.xlabel("Natural Log of Predicted Selling Price")
plt.ylabel("Frequency")
plt.title("2017 Predicted Selling Price with Reduced Permits Data")
plt.show()
y_pred = model.predict(X_test)
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(Y_test, y_pred))
print('MSE:', metrics.mean_squared_error(Y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(Y_test, y_pred)))
print('VarScore:',metrics.explained_variance_score(Y_test,y_pred))
# Visualizing Our predictions
fig = plt.figure(figsize=(10,5))
plt.scatter(Y_test,y_pred)
plt.xlabel("Natural Log of Actual Selling Price")
plt.ylabel("Natural Log of Predicted Selling Price")
# Perfect predictions
plt.plot(Y_test,Y_test,'r')
```

```
X = dataframe[["Year Built", "Year", "ln_prev_year", "Sqft Total"]]
Y = dataframe[dataframe.columns[7]]
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=101)
model_2 = Sequential()
model_2.add(Dense(40, input_dim=10, activation='relu', input_shape = (4,)))
model_2.add(Dense(30, activation='relu'))
model_2.add(Dense(20, activation='relu'))
model_2.add(Dense(10, activation='relu'))
model_2.add(Dense(1, activation = 'linear'))
model_2.compile(loss='mse', optimizer='Adam')
model_2.fit(X_train, Y_train, validation_data=(X_test, Y_test), epochs=150, batch_size = 32)
model_2.summary()
loss_df = pd.DataFrame(model_2.history.history)
loss_df.plot(figsize=(12,8))
y_pred = model_2.predict(X_test)
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(Y_test, y_pred))
print('MSE:', metrics.mean_squared_error(Y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(Y_test, y_pred)))
print('VarScore:',metrics.explained_variance_score(Y_test,y_pred))
# Visualizing Our predictions
fig = plt.figure(figsize=(10,5))
plt.scatter(Y_test,y_pred)
plt.xlabel("Natural Log of Actual Selling Price")
plt.ylabel("Natural Log of Predicted Selling Price")
# Perfect predictions
plt.plot(Y_test,Y_test,'r')
```

Reflection

While doing my project on quantifying the relationship between housing prices and permits in Austin, I learned how to code a neural network using Keras. I gained a better understanding of what a loss function is, and what an optimizer is. I plan on working in bioinformatics and these skills that I gained from this experience will help me code neural networks in the future. I also learned how to get the information needed from datasets. In this case, we had a permits dataset and I reduced that to the number of permits within a 2 mile radius for each home. I have never worked with datasets like this before and I learned how to extract the needed information.

My project contributed to the affordable housing project by Dr. Arya Farahi. While I did not work with anyone specifically on this project, I contributed to research on affordable housing in Austin by studying the impact of permits on housing value.

Overall, I had a wonderful experience working with Dr. Arya Farahi. He guided me through the process of reshaping datasets in a way that I have never learned before. He gave me the opportunity to learn how to code neural networks. I am grateful for the opportunity to work on the affordable housing project, given the rising cost of living in Austin.