House Price Prediction Report

Mr. Wong, MDM4U

Jerry Zhu

June 14th, 2022

Table Of Contents

1 Abstract

The real estate industry is a rich sector that introduces thousands of new investors, many that are looking to profit from the buying and selling of homes [1]. On the other hand, the act of selling a house at a fair price, and finding a suitable place to live, is an absolute necessity in today's society.
In the wake of the COVID-19 pandemic, the average house price in Toronto has gone up by 19.3% [2], and many brokers have been struggling to adjust to the sudden fluctuation in prices. Being able to predict house prices will help incoming sellers determine an acceptable selling price of a house and can help the customer find a residence that fits their budget. It will also transform and skyrocket the already popular real estate industry and unlock its true potential, and give buyers and homeowners the safety and security they desperately require in a heavily fluctuating market.
In this report, we will attempt to solve the fundamental problem of predicting the price of a house using its physical properties, including its condition, location, and features. Today, there is a large amount of data available on relevant statistics and contextual factors relating to house prices, in order to improve our understanding of the real estate industry. Notably, this problem has already been introduced and dissected in Zillow's Zestimate [3] and Kaggle's competition on housing prices [4].
The Ames Housing dataset is a comprehensive table of housing prices and corresponding features of homes in Ames, Iowa. Using this dataset from Kaggle, we will attempt to predict the price of a house using regression techniques, and further extend the accuracy of our hypothesis using a machine learning model and advanced regression [5]. Finally, using a web application [6], we will explore the practicality of such a model in real life, and its relevance in the current real estate industry.

2 Initial Analysis

First off, we have to find a quantitative variable to use as the dependent variable. To do this, we will extract the dataset, and perform an **exploratory data analysis**.

To achieve this, we import the Kaggle Dataset using the Kaggle API, setup a root folder to dump the zip archive file from the Kaggle competitions cloud, and extract the contents of the zip archive into the root folder. Finally, we delete the zip archive file, and view the contents of the dataset using the **data description** file: *data_ description.txt*.

2.1 Dataset Files

train.csv - Main dataset of aggregate values for each house.
test.csv - Another dataset of aggregate values for testing on smaller data.
sample_submission.csv - A sample of houses and their prices (used for predictions).
data_description.txt - A description of the compiled data.

2.2 Dependent Variable

From the description of the data content of the dataset, all the factors relate to housing price in some way, so `SalePrice` would be the most natural dependent variable.
We can also see that the data is cross sectional data, as the data is ID-stamped, not time stamped, and the data is collected on a single group of people, at some point in time.
To find a suitable independent variable to compare against the dependent variable, we can construct a correlation matrix, to determine the correlation of each variable with the dependent variable of `SalePrice`.
To do this, we will leverage Python and Google Colab's exploratory data analysis features, in order to analyze all relevant variables. We first convert the training dataset (which contain the aggregate entries) into a `DataFrame` using Python's Pandas module, to help us visualize the data. We will then drop all the non-numeric (qualitative) rows in the dataset, and create a correlation matrix using Pandas' `corr()` command [7]. Finally, we will sort the columns by the correlation, to find the quantitative variable(s) with the highest correlation with `SalePrice`, to be used as our independent variables.

2.2.1 Variables With Highest Correlation With SalePrice

	SalePrice
OverallQual	0.790982
GrLivArea	0.708624
GarageCars	0.640409
GarageArea	0.623431
TotalBsmtSF	0.613581

2.3 Independent Variable

From the correlation table, overall living quality `OverallQual` and size of living area `GrLivArea` have the highest correlations with `SalePrice`. We will use the size of living area (in square feet) `GrLivArea` as our independent variable to explain what our dependent variable depends on, since it has the least bias in its measurements, and we know with certainty how the data was collected.

This conclusion makes sense, as we know from prior knowledge that comfort of living and size of living area are important considerations when choosing a suitable house. However, we can extend on our prior knowledge, and use statistical analysis to explore exactly how crucial this variable is to the sale price of a house.

3 Data Collection

3.1 Hypothesis

My hypothesis is that as the size of living area (`GrLivArea`) increases, `SalePrice` increases as well.
To test this hypothesis, we will first isolate the data of the independent variable and the dependent variable, and graph the data to find such a correlation, and then prove this correlation.
To do this, we will drop all rows except the desired ones relating to the correlation we are trying to determine. We will then get a sample of the data, and graph it using Google Sheets, which will create a linear regression model including a scatter plot and line of best fit for the data. Using this, we can determine any obvious trends in the data.

3.2 Sampling Of Data

Since the dataset is very large (over 1000 rows), we will have to collect an unbiased sample of the data. To do this, we will use three common sampling methods: simple random sampling, systematic random sampling, and convenience sampling. For each sampling method, we will take a sample of exactly 100 rows. After collecting the samples using the various methods, we will save them as `.csv` (comma seperated values) files instead of dataframes, for easier access and storage. Finally, we will calculate and determine which sampling method is the best, using the correlation `corr()` of each sampling method.

3.3 Correlations of Different Sampling Techniques

3.3.1 Simple Random Sampling

	Id	GrLivArea	SalePrice
Id	1.000000	-0.044696	-0.055773
GrLivArea	-0.044696	1.000000	0.815114
SalePrice	-0.055773	0.815114	1.000000

3.3.2 Systematic Random Sampling

	Id	GrLivArea	SalePrice
Id	1.000000	0.015462	-0.029772
GrLivArea	0.015462	1.000000	0.742336
SalePrice	-0.029772	0.742336	1.000000

3.3.3 Convenience Sampling

	Id	GrLivArea	SalePrice
Id	1.000000	-0.012387	-0.162735
GrLivArea	-0.012387	1.000000	0.735129
SalePrice	-0.162735	0.735129	1.000000

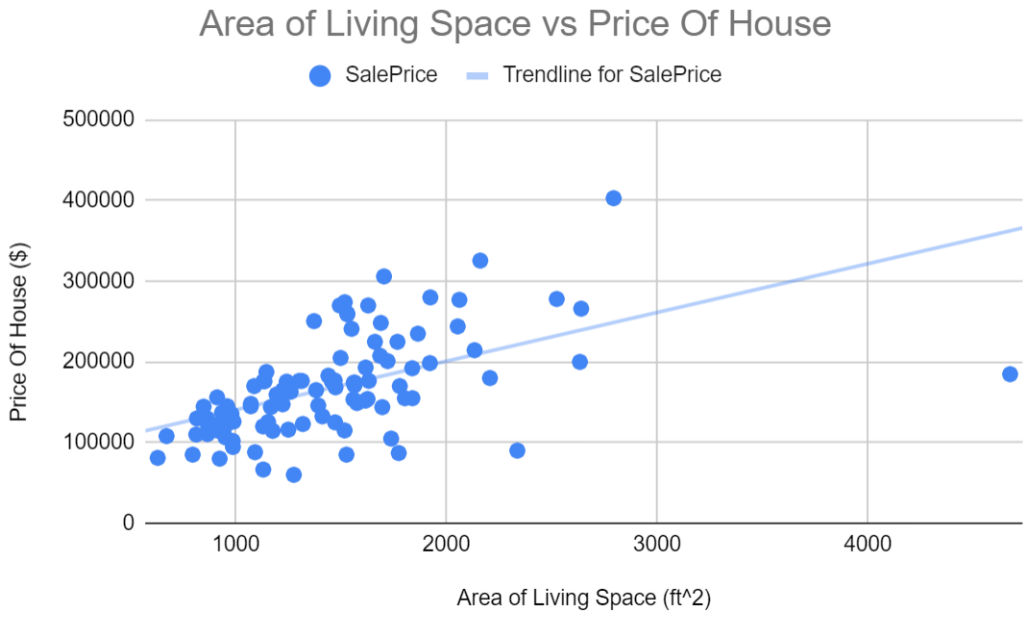
Next, we will compare the correlations of these sampling methods in order to determine the most accurate method.

4 Data Analysis

4.1 Sampling Conclusion

We notice that the best sampling method (the one that gives the highest correlations with `SalePrice`) is simple random sampling. This make sense because a purely pseudo-random sampling algorithm will generate the least amount of bias.
Next, we will graph the data of this sampling method to determine any trends in the two variables. We will graph a scatter plot and a line of best fit.
In order to decrease bias and keep the values in the sample dataset meaningful, we will keep the units of measurement, but adjust the scale accordingly.

4.2 Graphical Data



4.3 Trends And Conclusions

From the overall trend of the graph and the line of best fit, we see that there seems to be a strong positive correlation between the area of living space and the price of the house. Therefore, the variable we found is appropriate for the data, and we can continue by proving this correlation using a more in-depth statistical analysis of the trends and relationship of the variables, and try to predict the sale price using regression techniques.

5 Variable Analysis

Since we now know our dataset and sampling method, we will first extract this sample of 100 rows from the original dataset, and save it as a `.csv` file called `selected_data.csv`. We will then use Python functions to determine some general features of the data.

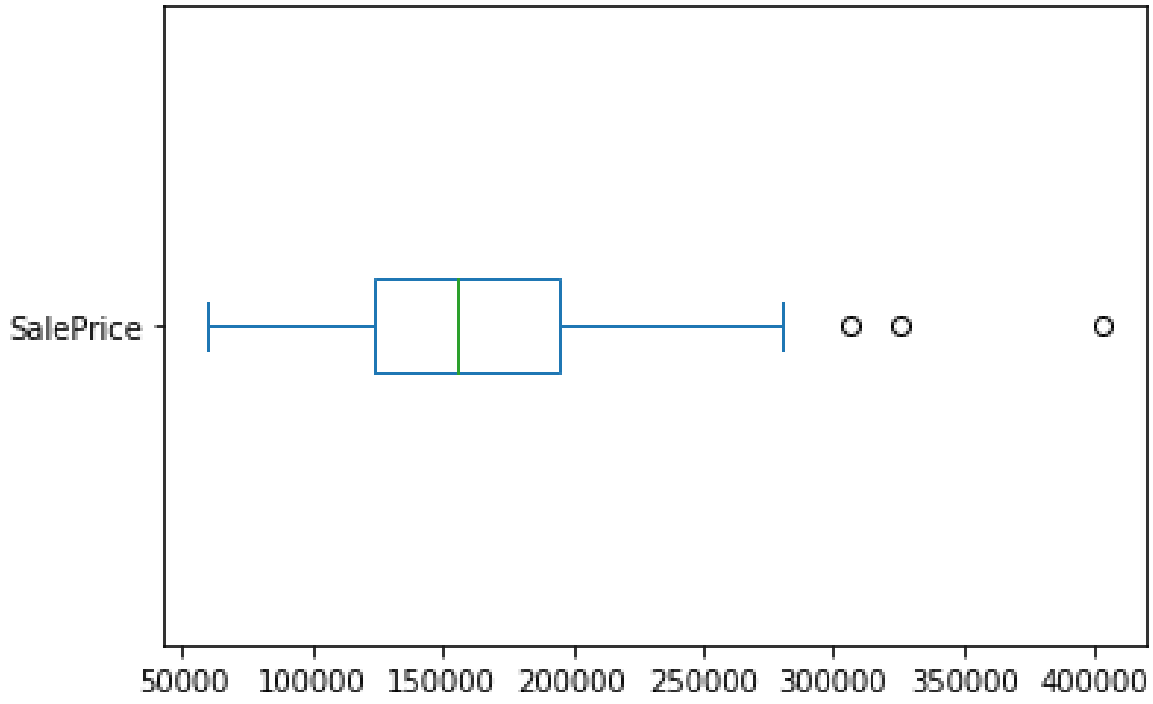
5.1 Sale Price Analysis

Mean: 167682.78
Median: 155500
Mode: None (all values are distinct)
Variance: 3860487358.56
Standard deviation: 62132.82

5.1.1 Quartiles

First quartile: 124125
Second quartile (median): 155500
Third quartile: 194375
Interquartile range: 70250

5.1.2 Box and Whisker Plot



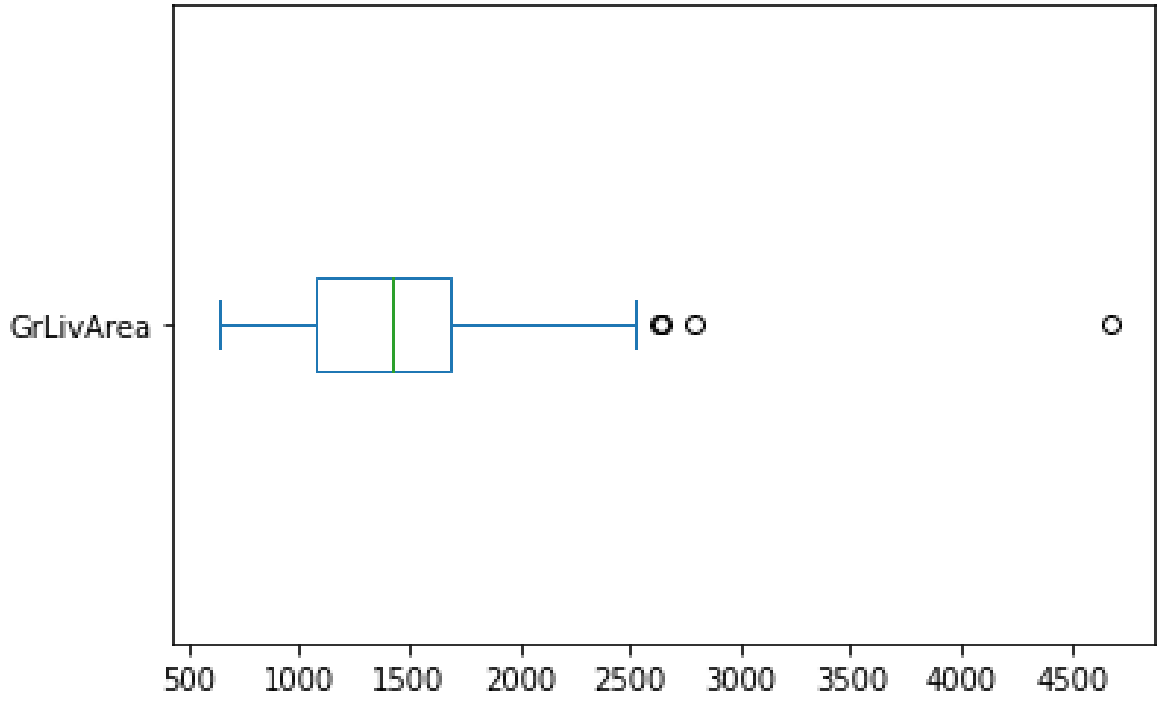
5.2 Living Area Analysis

Mean: 1450.01
Median: 1426
Mode: None (all values are distinct)
Variance: 311012.05
Standard deviation: 557.68

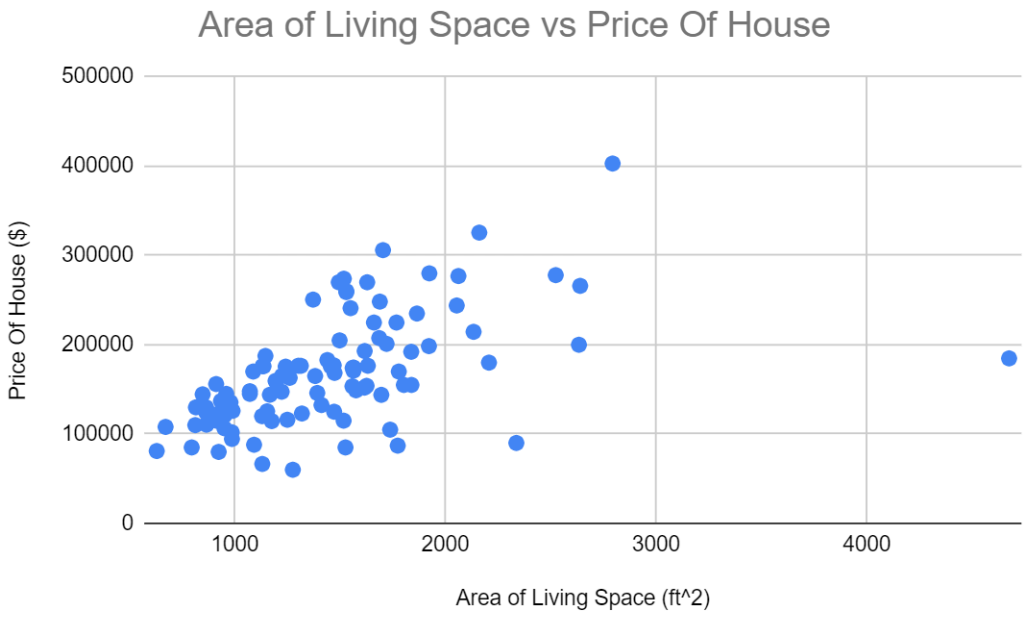
5.2.1 Quartiles

First quartile: 1072
Second quartile (median): 1426
Third quartile: 1686.75
Interquartile range: 614.75

5.2.2 Box and Whisker Plot



5.3 Graphs Of Data



5.4 Regression Techniques

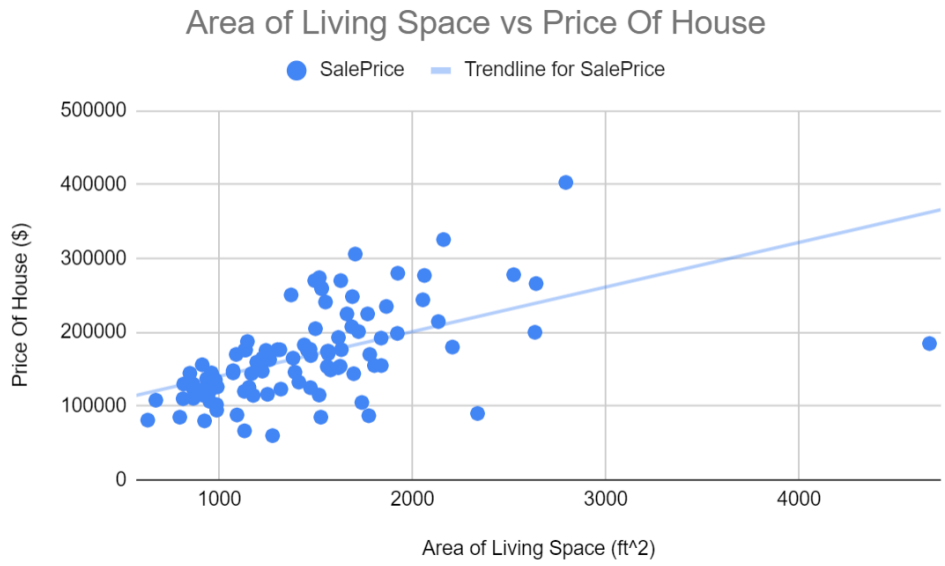
Given this data, we can leverage regression techniques to try to interpolate and extrapolate the sale price based on the living area. We will also check how accurate this data is compared to actual sale price values.

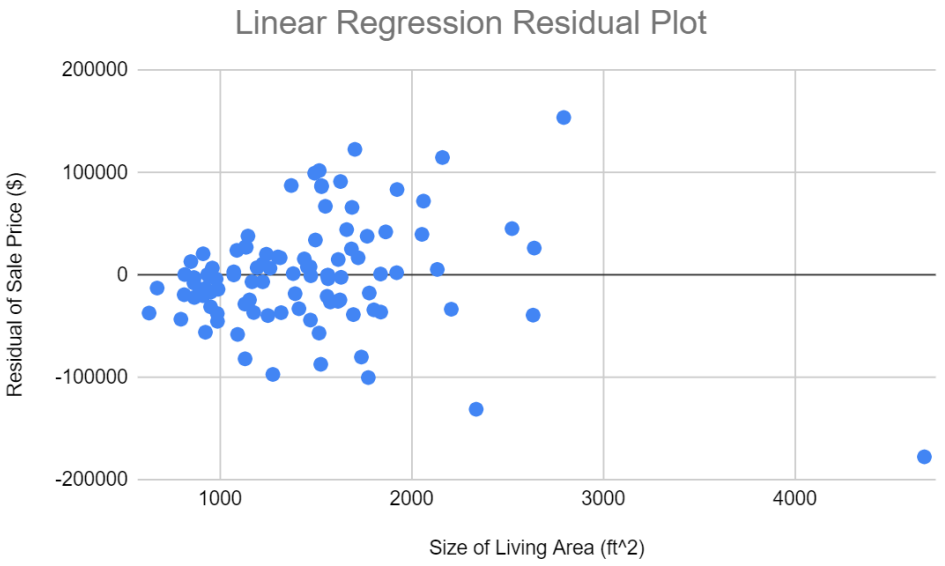
6 Regression Analysis

We can perform different regressions on the data, and check their correlations to the sample data. To do this, we will generate a line or curve of best fit, and find the correlation coefficient of that data, for each type of regression.

6.1 Linear Regression

$y_1 \sim ax_1 + b$
a = 60.40067407
b = 80101.1986
Equation: y = 60.40067407x + 80101.1986
$r^2 = 0.294$

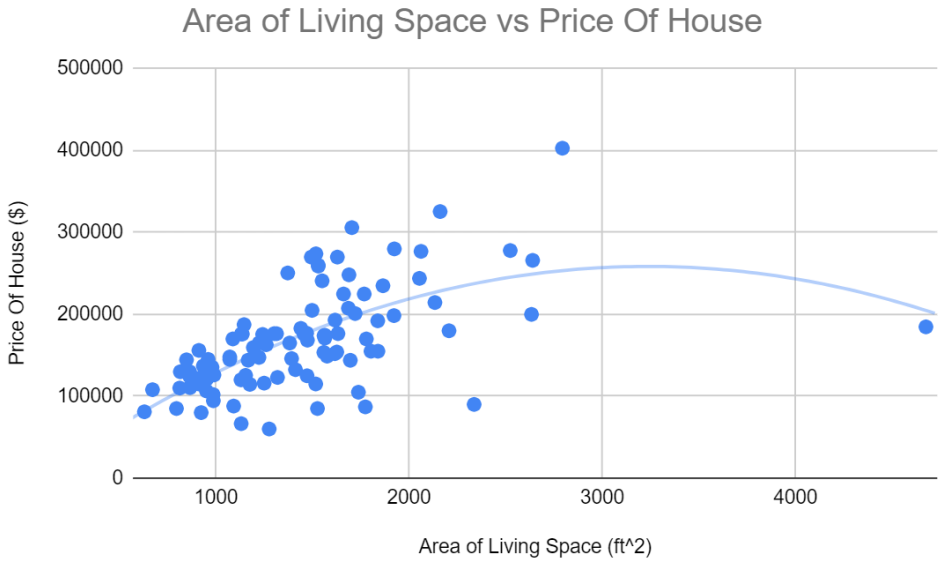


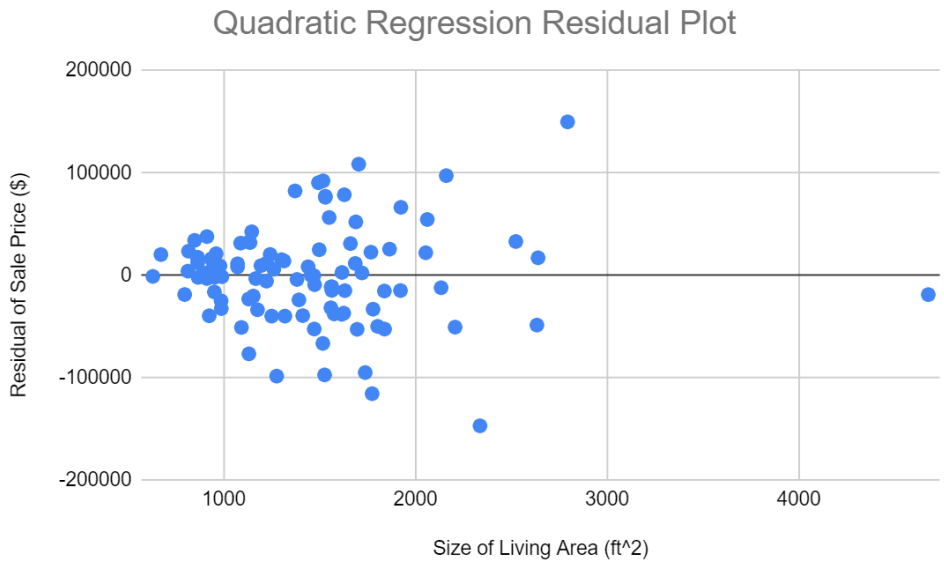


6.2 Non-Linear Regression

6.2.1 Quadratic regression

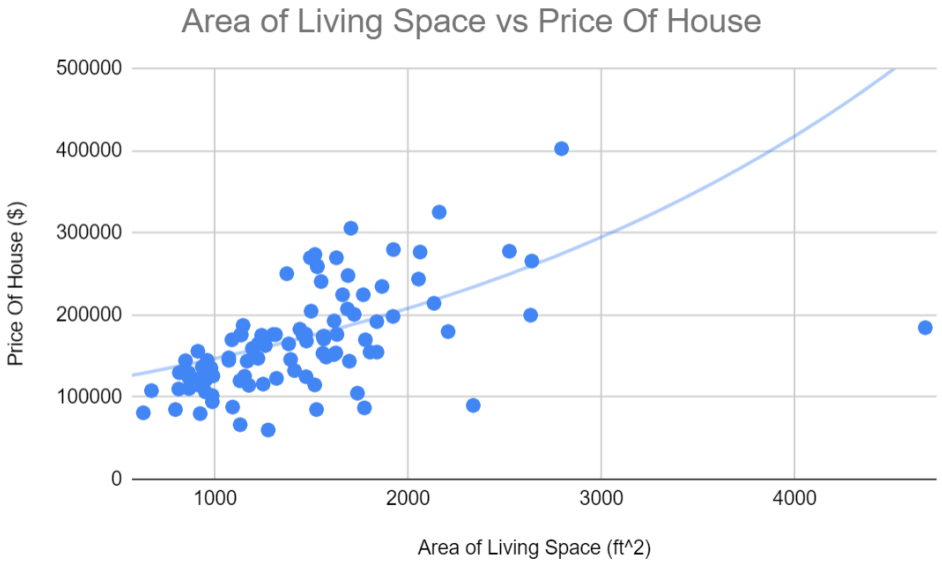
$y_1 \sim a{x_1}^2 + \mathrm{b}x_1 + \mathrm{c}$
a = -0.026
b = 168
c = -13459
Equation: y = -13459 + 168x + $-0.026x^2$
$R^2 = 0.407$





6.2.2 Exponential Regression

$y_1 \sim ae^{bx_1}$
a = 103852
b = 0.000348
Equation: y = 103852*$e^{0.000348x}$
R^2 = 0.067



From the regressions observed, we can conclude that the quadratic regression is the best fit for the data, as it has the highest R^2 (or r^2) value. However, since there are many other extraneous factors in determining the price of a house, solely the size of living space may not be the best method of determining the price of a house. In the next section, we will leverage all features in order to perform a less biased regression on the data.

7 Machine Learning

We notice that our regressions, although visually accurate, do not give good correlation coefficient (R^2 values). Thus, we can use machine learning models to optimize the model to get better correlations to interpolate or extrapolate the data more accurately. To do this, we can use various pretrained machine learning models, that apply multilinear regression.
A machine learning model works by taking a set of data with certain features or hyperparameters, and tuning them to clean the data. After that, the model will use these tuned features and generate a set of weights that acts as a function from the independent variable (living area) to the dependent variable (sale price). However, since these models are trained to fit nearly perfectly to a dataset, using correlation coefficients is redundant. We can instead calculate the accuracy of this model using an metric known as RMSLE (Root Mean Squared Logaritmic Error), calculated by this equation:

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N}(log(p_i+1)-log(a_i+1)^2)}$$

where a_i is the actual price of the house and p_i is the predicted price of the house. This metric is easy to understand as it gives us an easy comparison between models: the smaller the RMSLE, the smaller the error between the predicted and actual values, and the better the model is. This value will be more useful than the correlation coefficient in terms of comparing to find the best model. We will train our data (`train.csv`) on a machine learning model. First, we must use all the data, in order to give more datapoints for the model to interpret. We must then clean the data, by extracting and dropping any columns that are unnecessary, and converting the data from categorical (text) to numerical (numbers), in order to give the model an easier time interpreting the features using a `LabelEncoder`. By doing this, we tune the columns of data into hyperparameters that can generate a more accurate model. Finally, we use Python's `scipy` module to apply a box cox transformation on the data, to get rid of any skewness in the data distribution.
Now that our data is properly processed, we will now train our machine learning model. We first need to generate folds or epochs, for how many iterations the model will go through. Then, we need to define the RMSLE as a cross validation score (a success metric of accuracy), and run it against our model in order to determine if it was successful in predicting the sale price. Below, many pretrained models were tested using Python and Google Colab, as well as the Scikit machine learning library, and the cross validation score was tested and compared in order to determine the best model for sale price regression.

7.1 Model Information

7.1.1 Random Forest Regression

The simplest (and go to regression method) is a random forest regression model. It generates a randomized decision tree, and changes the weights of the trees at each point. This random "forest" acts as a meta estimator that uses various splits of the dataset to fit the decision tree and uses pseudorandom averaging to improve the predictive accuracy and prevent inaccuracies such as overfitting.
Model Used: `forest_model = RandomForestRegressor(max_depth=2, random_state=7)`
RMSLE score: 0.2339

7.1.2 LASSO Regression

A LASSO (least absolute shrinkage and selection operator) model is a regression analysis model that uses regularization to enhance prediction accuracy, and by making the features easier to interpret, improve the prediction metric of the model.
Model Used: `lasso_model = make_pipeline(RobustScaler(),`
`Lasso(alpha=0.0005, random_state=1))`
RMSLE score: 0.1125

7.1.3 ElasticNet and Kernel Ridge Regression

Kernel ridge regression (KRR) combines ridge regression analysis and regularization with a kernel trick optimization, by generating a linear function in the space created by a kernel matrix of the data. The elastic net is a regularized regression method that linearly combines the advantages and penalties of the LASSO and kernel ridge methods. Both these methods are similar to the LASSO regression, and will be extremely useful in later steps.
Model Used: `enet_model = make_pipeline(RobustScaler(),`
`ElasticNet(alpha=0.0005, l1_ratio=0.9, random_state=3))`
RMSLE score: 0.1126

Model Used: `kri_model = KernelRidge(alpha=0.6,`
`kernel='polynomial', degree=2, coef0=2.5)`
RMSLE score: 0.1150

7.1.4 XGBoost

XGBoost is a gradient boosting algorithm to boost accuracy, and improve regression analysis by leveraging performance based decision trees.
Model Used: `xgb_model = xgb.XGBRegressor(colsample_bytree=0.4603, gamma=0.0468, learning_rate=0.05, max_depth=3, min_child_weight=1.7817, n_estimators=2200, reg_alpha=0.4640, reg_lambda=0.8571, subsample=0.5213, silent=1, random_state=7, nthread=-1)`
RMSLE score: 0.1155

7.1.5 Light GBM

Light GBM is a gradient boosting algorithm, similar to XGBoost.
Model Used: `lgb_model = lgb.LGBMRegressor(objective='regression',num_leaves=5, learning_rate=0.05, n_estimators=720, max_bin=55, bagging_fraction=0.8, bagging_freq=5, feature_fraction=0.2319, feature_fraction_seed=9, bagging_seed=9, min_data_in_leaf=6, min_sum_hessian_in_leaf=11)`
RMSLE score: 0.1157

From these models, we can see that the best singular model is the LASSO model, but all models are fairly close in their RMSLE scores. To significantly improve the RMSLE scores of our machine learning model, we will use a method known as stacking.

7.1.6 Averaged Model Stacking

Our first method of stacking, is to take many different models and stack them together by averaging them using a weighing function. By averaging the LASSO, XGBoost and Light GBM models (the latter two used for gradient boosting to augment the accuracy of the model), we can create a model with a significantly lower RMSLE.
Model Used: `averaged_model = AveragingModels(models = (lasso_model, xgb_model, lgb_model))`
RMSLE score: 0.1100

7.1.7 Meta Model Stacking

Finally, we can go one step further. By using a meta model (the LASSO regression model), we are able to further improve our model, by using the LASSO model as a "basis" for other models to add on and tune. This will highlight our best single model, while utilizing the benefits of other models as well.
Model Used: `stacked_models = StackingAveragedModels(base_models = (enet_model, xgb_model kri_model), meta_model = lasso_model)`
RMSLE score: 0.1093

7.1.8 Ensembling

The last step we can leverage is called ensembling, which can further increase the accuracy of the model, by adjusting the weights of the stacked meta model, as well as fully optimizing XGBoost and LightGBM gradient boosting. Since this method is much more time consuming, we will simply mention the final product and intuition behind it.

```
ensemble = stacked_pred*0.50 + xgb_pred*0.25 + lgb_pred*0.25
```

RMSLE score: 0.1009

This is the lowest score we can get to using machine learning models and the Ames dataset, so we can use the ensembling method to predict the sale price of a house based on the features of our dataset. To see the exact results of this machine learning model, a file called `submission.csv`, containing predictions of the sale price of a house based on its features, has been generated based on the results of the testing dataset `test.csv`.

To see an interactive web application with all the regressions summarized and visualized, go to housepriceprediction.ml.

8 Conclusion

From the analysis and regression of the data, we can tell that there is a strong, positive correlation between the area of living space of a house, and the price of it. This also demonstrates that the correlation is by cause and effect, since the more area a house has, the more appealing it is, and the more people are willing to pay to buy it.
Furthermore, the strongest correlation, the quadratic correlation can also be explained. As the area of a house increases to a certain point, the sheer price of the house becomes discouraging, even with the added living space, as it becomes less useful (utility of extra space decreases), and thus the price must increase less, or decrease, to keep the house appealing to buy. We could also collect more data, to create a better curve of best fit (like a logarithmic curve), in order to get a higher correlation. As the demand for very large houses decreases, the supply also drops, thus changing the price, which is why we see a quadratic regression fits the model the best, as compared to an linear or exponential regression.

8.1 Error

There were many possible cases of errors and biases, but none that drastically altered the results of the data in any way. There were biases in sampling methods, so we picked the method that would give us the least amount of bias. There were possible biases in data collection, so we used the most comprehensive dataset of house prices in Iowa, in order to poll from a hopefully more varied sample of data. There could have been outliers and errors as well in the sample data (`selected_data.csv`), but we made sure that the sample data matched the spread and tendencies of the dataset, as to minimize the amount of bias in sampling. Finally, we explored any assumptions that could have been made, and made sure to clean the data during machine learning, and discussed the use of categorical and numerical variables. While converting and unskewing the data, we made sure that any "objective" data, like overal quality, or hard to understand categorcial data was either removed, or fitted to a more useable column, in order to remove any biases and limitations in the prediction of the model based on the dataset. All in all, any errors in the dataset were mitigated quite well, as our prediction model (both basic and advance regression) produced high accuracies.

8.2 Comparisons

We can also compare the values of the data set (housing data in Ames, Iowa) with housing data in all of America, to see how accurate this data is. We can also determine if all American households can benefit from the results of this report.
From the sample data `selected_data.csv`, we can conclude that the average house price is about 1450.01 m^2, and the average sale price is about \$167682.78. From the entire dataset `train.csv`, the average house price is about 1515.46 m^2, and the average sale price is about \$180921.19. Since the values are relatively close together, we can reasonably establish that the sample we chose was an unbiased and good sample of the dataset. When we compare this data to statistics data from the all of America, we see that houses in Ames, Iowa are cheaper than the median. In fact, according to the Federeal Reserve Bank economic data of St.Louis [8], the average price for houses in the 4th quarter of 2011 was \$259700. Since the average price of the dataset is still fairly close to the average house price in America, all Americans could benefit from the findings of this report. If we wanted to tune the model even more, we could collect more data, this time from other states as well, for example using stratified sampling, to get a complete sample set of data of housing data for all American homeowners.

8.3 Summary

The issue of predicting house prices is a very complex one, that requires immense amounts of data, cleaning, modeling, and prediction. Through this report, we have highlighed and analyzed some of the ways we can try to predict the price of a house based on certain numerical or categorical features. As we start to gain a better understanding of regression through learning data management, we can understand exactly how these methods work, and gain insight into other algorithms that will help us predict a lot more than just house prices, some that we may even come up with on our own. For example, a more modern model called Hedonic Regression [9] may work on problems of this type, although research into the advantages of this model are still ongoing. Finally, it's important for us to discuss house price prediction because of its utility value. It can help us make informed decisions on the actual price of a house, and reduce any human bias from the housing market. As we learn more about house prices, we can gain an appreciation of the ups and downs of the real estate industry, and what it takes to thrive in a constantly changing world.

9 Works Cited

[1] https://www.researchgate.net/publication/320801620_Modeling_House_Price_Prediction_using_Regression_Analysis_and_Particle_Swarm_Optimization_Case_Study_Malang_East_Java_Indonesia

[2] https://globalnews.ca/news/8400321/canada-housing-prices-central-bank-warning/

[3] https://www.zillow.com/blog/zestimate-updates-230614/

[4] https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data

[5] https://m2pi.ca/project/2020/bc-financial-services-authority/BCFSA-final.pdf

[6] http://housepricepredictionml/

[7] https://pandas.pydata.org/docs/reference/index.html

[8] https://fred.stlouisfed.org/series/ASPUS

[9] https://www.investopedia.com/terms/h/hedonic-regression.asp

9.1 Appendix

To see the collected data, go to selected$_d$*ata.csv* To see the repository where all the relevant files and data are generated, see the Github repository.
To see the interactive website, go to housepricepredictionml and view the site.