10.2 Course Project: Milestone 5 – Final Project Paper

Keiuntae Smith

DSC 630: Predictive Analytics

Dr. Fadi Alsaleem

August 11, 2022

Problem Statement

House Price prediction is an important element to drive Real Estate efficiency. For the purposes of this predictive analysis project, I would like to examine if there is a dependable process through machine learning to predict the house sale price and what variables are important during this data analysis process. In simpler terms, I want to generate a formula on specific features that will provide me the price or estimate price of the house.

Intended Audience and Importance

During earlier times, House prices were determined by calculating the acquiring and selling price in a locality. A House Price prediction model could be extremely beneficial for filling in the information gap and improve Real Estate efficiency. "An accurate forecast on the house price is important to prospective homeowners, appraisers, developers, tax assessors, investors, and other real estate's market participants, such as mortgage lenders and insurers" (Begum, Kheya, & Rahman, 2022). This can be an important algorithm for homebuyers and home developers to take advantage of. Homebuyers will be able to arrange the right time to purchase a house and developers can have a better way to determine the selling price of a house. It could assist in eliminating some of the guess work involved in the real estate industry. Overall the algorithm could help Investments and Residential teams make data driven decisions.

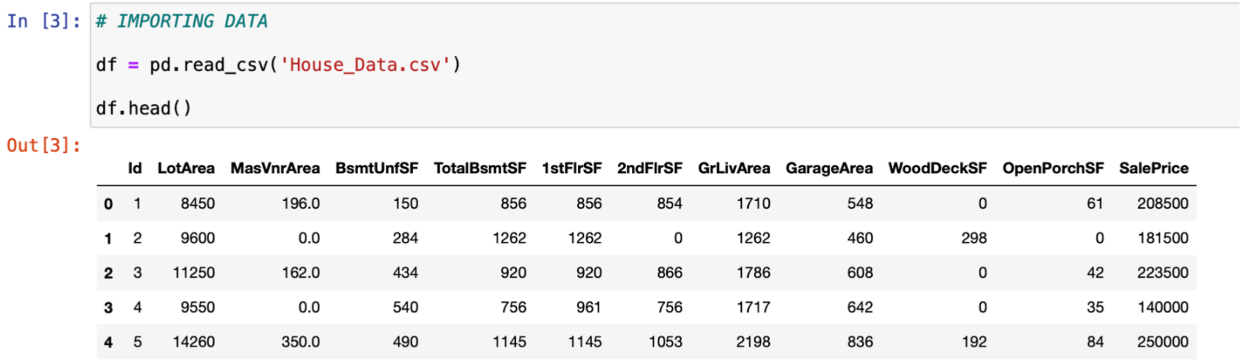
Dataset Details

I will be using a housing dataset from Kaggle that examines data from various cities with several attributes. The attributes included for my analysis is as follows:

- ID
- Date
- Masonry Veneer Area

- Basement Unfinished Square Foot
- Total Basement Square Foot
- 1st Floor Square Foot
- 2nd Floor Square Foot
- Above Ground living Area
- Garage Area
- Wood Deck Square Foot
- Open Porch Square Foot
- Sale Price

Below is the Python code snapshot of the dataframe I will be using for my analysis.



```
In [3]: # IMPORTING DATA

        df = pd.read_csv('House_Data.csv')

        df.head()

Out[3]:
```

	Id	LotArea	MasVnrArea	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	GrLivArea	GarageArea	WoodDeckSF	OpenPorchSF	SalePrice
0	1	8450	196.0	150	856	856	854	1710	548	0	61	208500
1	2	9600	0.0	284	1262	1262	0	1262	460	298	0	181500
2	3	11250	162.0	434	920	920	866	1786	608	0	42	223500
3	4	9550	0.0	540	756	961	756	1717	642	0	35	140000
4	5	14260	350.0	490	1145	1145	1053	2198	836	192	84	250000

Model Choice

Based on the type of prediction and outcome I am attempting to achieve I know I need to choose a regression model. The ultimate goal here is to predict a numerical outcome, which in my case a sale price. The main model I intend to apply to my analysis is the linear regression. The Linear regression is a type of model where the relationship between an independent variable and a dependent variable is assumed to be linear. It is a simple statistical regression method used for predictive analysis and shows the relationship between the continuous variables. I think this will work hand in hand with the dataset chosen.

Results Evaluation Plan

I plan on using the following metrics to evaluate the results generated: explained variance score metric and the r2 squared metric. Both of these metric functions are provided by the Scikit-learn package that python offers. The explained variance score will explain the dispersion of errors of the dataset I chose to use. The R squared is a score that measures how well the dependent variable of my dataset explains the variance of the independent variable. This metric is commonly used and accepted for most regression models.

During this analysis, I hope to find out if the models I've selected will actually achieve an explained variance score above 70 percent once applied. I also am hoping to learn, if possible, what variables of the dataset that are particularly correlated to the housing prices, negatively or positively.

Proposal Risks

One of the risks that I am concerned with of my proposal is overfitting. Logistic regression models attempt to predict outcomes based on a set of independent variables, but they can be vulnerable to overconfidence. This means that my model can appear to possess more predictive power than it actually has. If the model is generating too much noise, an overfit model will consequently make predictions based on that noise. Overfitting can single-handedly ruin the machine learning model.

Contingency Plan

As a contingency plan, I have 3 other housing datasets with different parameters and variables if this dataset does not give me the desired results. Also, if I don’t achieve optimal results from the model I chose, I would attempt a different model to include the Decision tree and random forest models.

Data Preparation and Exploratory Data Analysis

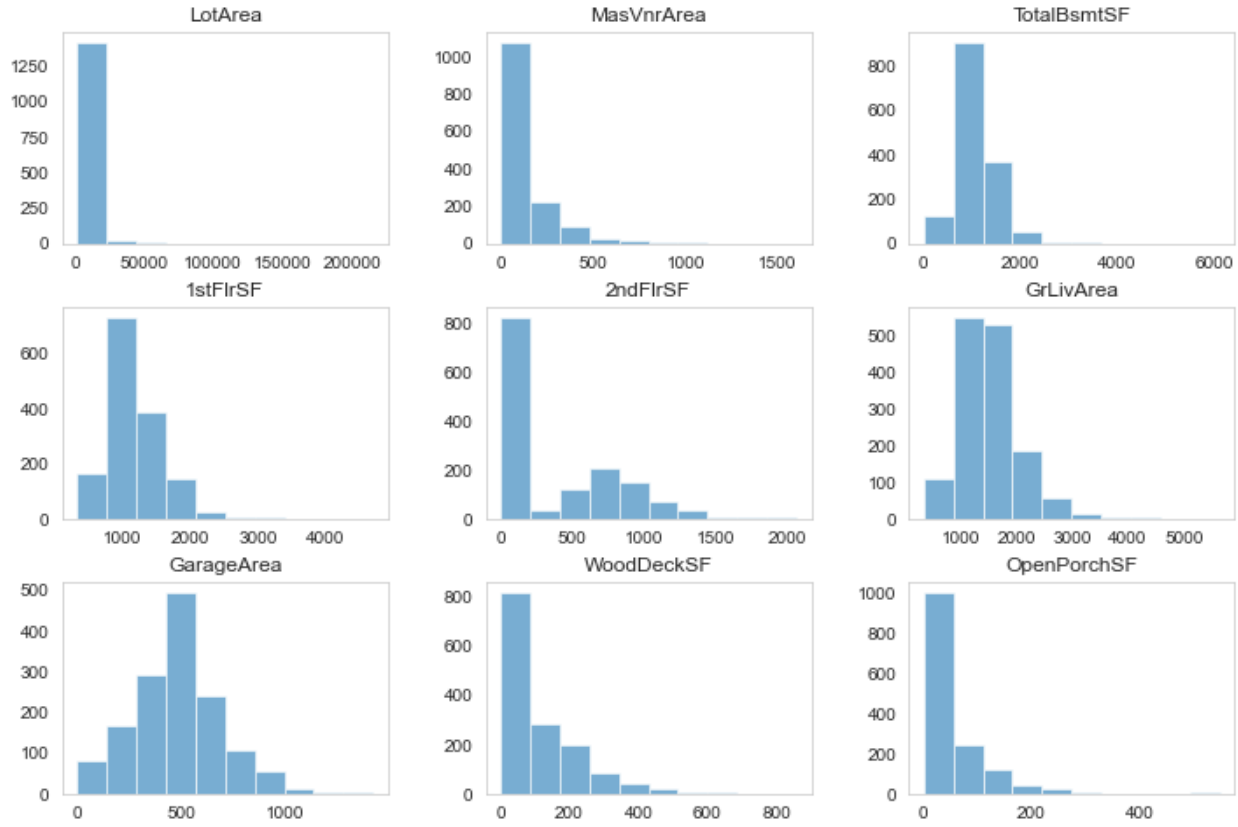
To prepare this dataset, I had to transform the dataframe a bit to make it more acceptable. The first step in conducting this transformation was to isolate the ID and set is as the index. Next, I had to remove all the null values contained in the dataset by using the dropna function. This action was followed by identifying any variables that were not of integer type. After examining the dataframe, using the info() function, I identified one variable that was “float 64”. I converted that variable to an integer to mirror the other variables. See the Python code below for details:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1452 entries, 1 to 1460
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   LotArea      1452 non-null   int64
 1   MasVnrArea   1452 non-null   int64
 2   BsmtUnfSF    1452 non-null   int64
 3   TotalBsmtSF  1452 non-null   int64
 4   1stFlrSF     1452 non-null   int64
 5   2ndFlrSF     1452 non-null   int64
 6   GrLivArea    1452 non-null   int64
 7   GarageArea   1452 non-null   int64
 8   WoodDeckSF   1452 non-null   int64
 9   OpenPorchSF  1452 non-null   int64
 10  SalePrice    1452 non-null   int64
dtypes: int64(11)
memory usage: 136.1 KB
```

Once the data transformations were achieved, I began by doing some preliminary exploratory data analysis. I wanted to do this by capturing the data in a visual way by using 3 different graphs: figure 1 (multiple histograms), figure 2 (Heatmap), and figure 3 (distribution plot).

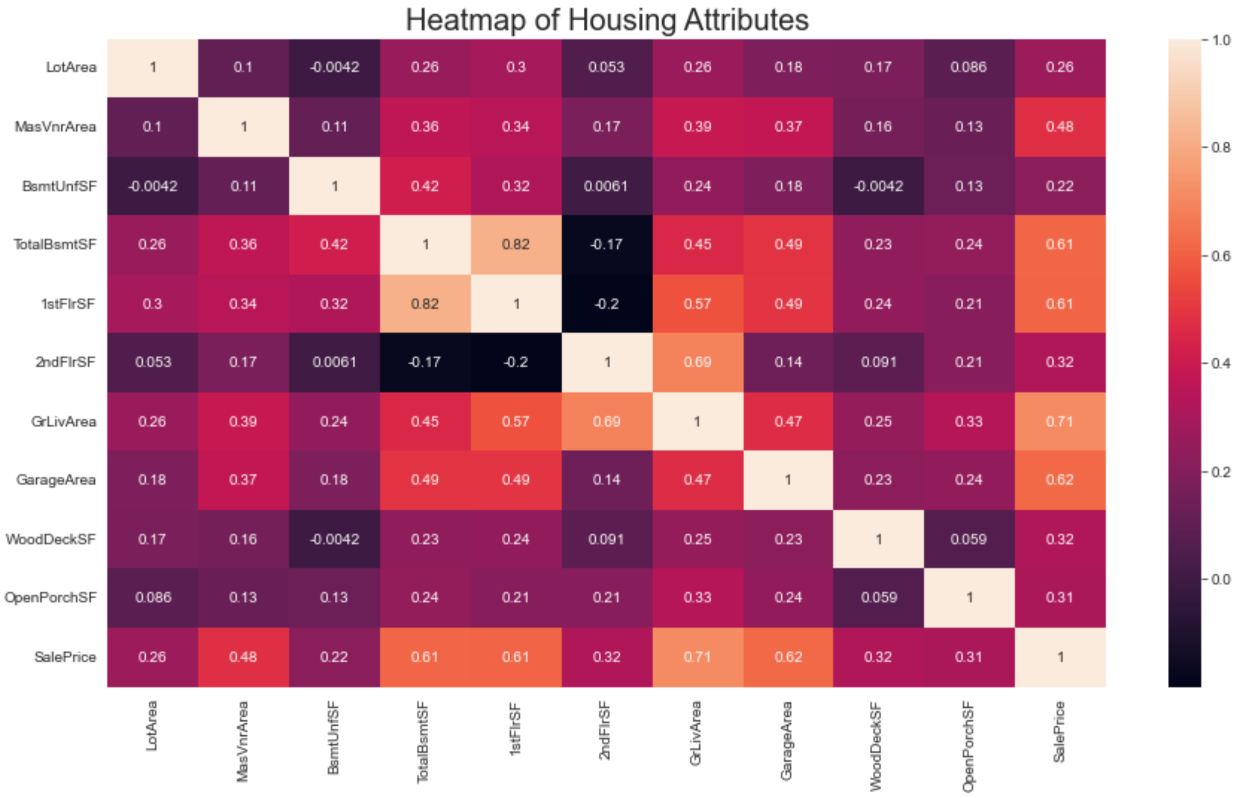
I began by generating histograms for the selected numerical features. They give a rough sense of the density of the underlying distribution of the housing data.

Figure 1:



Following the histograms, I produced a correlation heatmap to illustrate relations between two variables within the dataset.

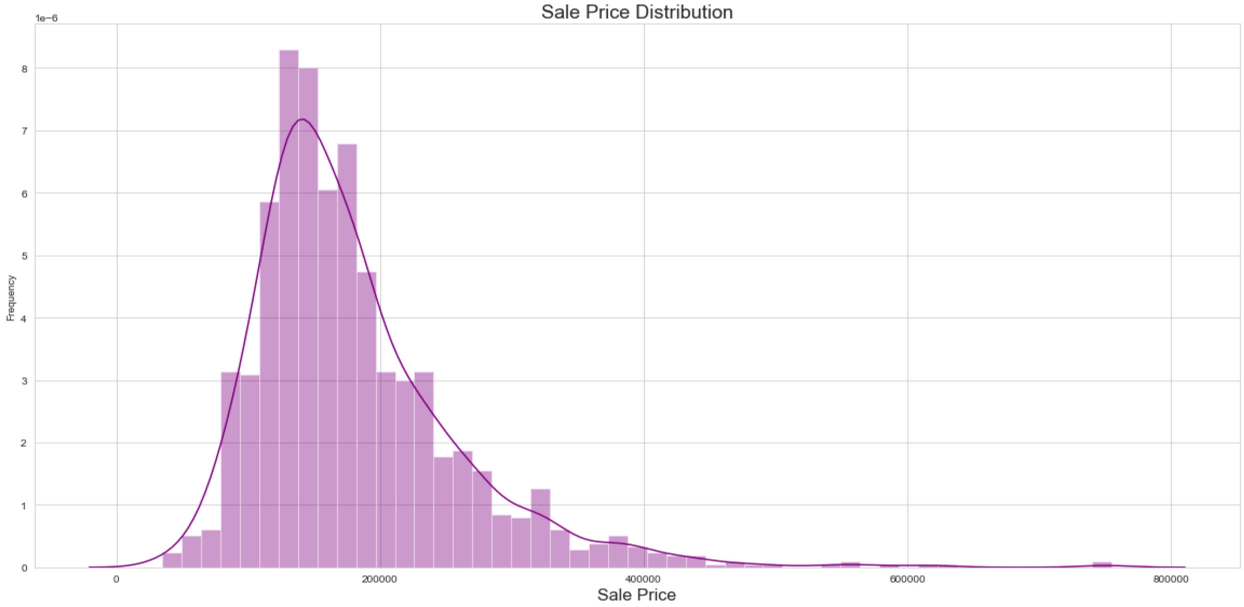
Figure 2:



	LotArea	MasVnrArea	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	GrLivArea	GarageArea	WoodDeckSF	OpenPorchSF	SalePrice
LotArea	1	0.1	-0.0042	0.26	0.3	0.053	0.26	0.18	0.17	0.086	0.26
MasVnrArea	0.1	1	0.11	0.36	0.34	0.17	0.39	0.37	0.16	0.13	0.48
BsmtUnfSF	-0.0042	0.11	1	0.42	0.32	0.0061	0.24	0.18	-0.0042	0.13	0.22
TotalBsmtSF	0.26	0.36	0.42	1	0.82	-0.17	0.45	0.49	0.23	0.24	0.61
1stFlrSF	0.3	0.34	0.32	0.82	1	-0.2	0.57	0.49	0.24	0.21	0.61
2ndFlrSF	0.053	0.17	0.0061	-0.17	-0.2	1	0.69	0.14	0.091	0.21	0.32
GrLivArea	0.26	0.39	0.24	0.45	0.57	0.69	1	0.47	0.25	0.33	0.71
GarageArea	0.18	0.37	0.18	0.49	0.49	0.14	0.47	1	0.23	0.24	0.62
WoodDeckSF	0.17	0.16	-0.0042	0.23	0.24	0.091	0.25	0.23	1	0.059	0.32
OpenPorchSF	0.086	0.13	0.13	0.24	0.21	0.21	0.33	0.24	0.059	1	0.31
SalePrice	0.26	0.48	0.22	0.61	0.61	0.32	0.71	0.62	0.32	0.31	1

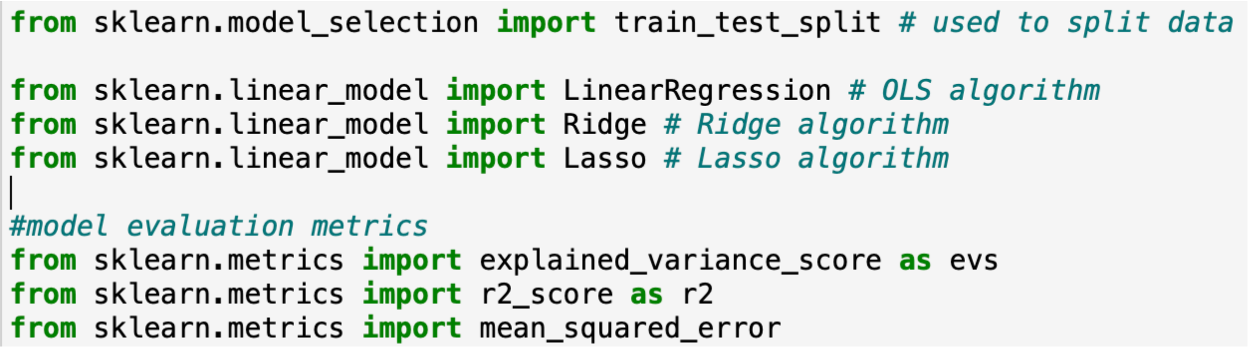
Finally, I manufactured a distribution plot of the Sale Price. This plot will help visualize the distribution of the Sale Price by combining the functionality from a histogram, rug plot, and a kde plot.

Figure 3:



Data Modeling and Evaluations

During the build a model phase of this milestone, I chose to use linear regression models because they have become the proven way to scientifically and reliably predict the future. The three linear algorithms I preferred were Ordinal Least Square, Lasso, and Ridge. I chose three so that I can compare the three to each other for superiority. The Python libraries are as followed:



```
from sklearn.model_selection import train_test_split # used to split data

from sklearn.linear_model import LinearRegression # OLS algorithm
from sklearn.linear_model import Ridge # Ridge algorithm
from sklearn.linear_model import Lasso # Lasso algorithm

#model evaluation metrics
from sklearn.metrics import explained_variance_score as evs
from sklearn.metrics import r2_score as r2
from sklearn.metrics import mean_squared_error
```

Next, I had to split the data into a training and test set. The X and y variables were defined and were consequently split into train and test by using the train_test_split function. At this point, a model is ready to be built and evaluated. I started with Ordinal Least Square algorithm and achieved a coefficient of determination (r2) of .772383636249033 and an explained variance score of .7733369645921897. Lasso Regression model reached a coefficient of determination (r2) of .7723836623441476 and an explained variance score of .7733370054270638. Finally, the Ridge algorithm accomplished a coefficient of determination (r2) of .7723836360889462 and an explained variance score of .7733369645976185. To my surprise all three algorithms yield pretty much the same results. This communicates to me that all the models I chose performed fairly well on this particular dataset.

Lessons Learned

What I have learned through this analysis project is that a good exploratory data analysis is necessary before implementing a model. Exploratory Data Analysis does two main things: aids in the clean-up of a dataset and it gives a better understanding of the variables and the relationships between them (Shin, 2019). In addition, Data pre-processing and feature engineering can be beneficial in transforming data for a smooth implementation. Lastly, Linear regression is expeditious, interpretable and well understood but it may not be the elitist predictive accuracy due to the fact that it assumes a linear relationship between the inputs and the target. Based on the results from my analysis, I do not believe that my model is totally ready for deployment. I do; however, have faith that it is close to being ready do to the great start of this analysis.

Recommendations and Future Work

- Use regularization to keep all the attributes from the dataset and reduce the magnitude of the parameters.

- Use the cross-validation instead of the train/test method to overcome high variance. “This method results in a less biased model compared to other methods since every observation has the chance of appearing in both train and test sets” (Goyal, 21).
- Improve the selected model by tweaking other parameters to improve the performance metrics.
- Test the model on multiple datasets to see how it performs on other data.

References

Begum, A., Kheya, N. J., & Rahman, Z. (2022, January). *Housing Price Prediction with Machine Learning*. Retrieved from https://www.ijitee.org/wp-content/uploads/papers/v11i3/C97410111322.pdf

Goyal, C. (21, May 2021). *Importance of Cross Validation: Are Evaluation Metrics enough?* Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/05/importance-of-cross-validation-are-evaluation-metrics-enough/

Shin, T. (2019, December 12). *Exploratory Data Analysis — What is it and why is it so important? (Part 1/2)*. Retrieved from medium.com: https://medium.com/swlh/exploratory-data-analysis-what-is-it-and-why-is-it-so-important-part-1-2-240d58a89695