Predicting Housing Prices Using

Advanced Regression Techniques

Jordan Matthews

Zachary Golla

Member Contribution Statement:

All work was done by team members in unison so that the amount of work done by each team member would be equal. Because of this, each team member contributed to 50% of the total work time used to complete the project. The details of each team member's work completed during this time will be revealed throughout this paper.

Choosing a Project (Both):

To choose the project we looked through the different Kaggle competitions. Specifically we looked through the currently open and available competitions. There were many, but we felt this was both interesting and was a great example of machine learning being used in a real world scenario.

Planning (Both):

To begin we picked one regression algorithm, linear regression. We decided to first read in and look at the data that we were dealing with. We realized very quickly that the data sets were very different from each other in many ways. We decided that before running the algorithm we had chosen, we should first try and understand in what ways we should preprocess the data. It was concluded that once the data was preprocessed, we would have a better understanding of how the data will apply to our regression algorithms.

Preprocessing the Data

There are 79 types of data provided to help predict the sales price, the data set includes data that is nominal qualitative, ordinal qualitative, and discrete quantitative. Each of the 79 data types and their types are listed below.

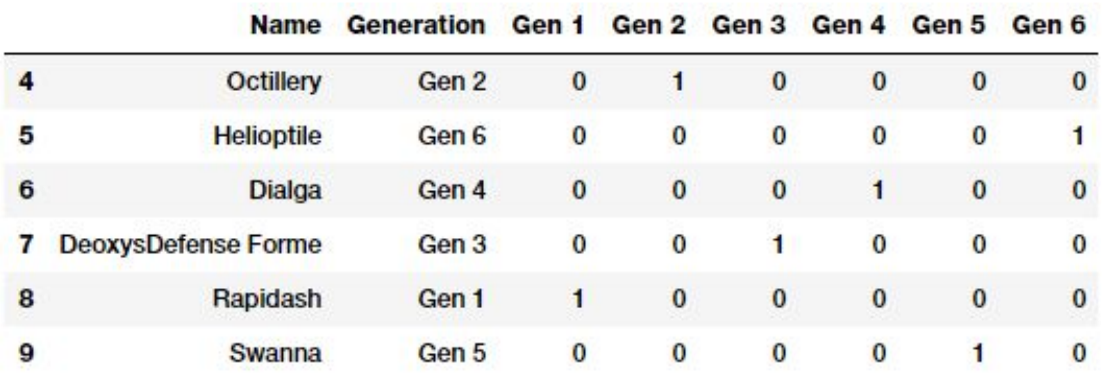
Type of Data	Name of Data
Nominal Qualitative	MSZoning, Utilities, LotConfig, LandContour, Neighborhood, Condition1, Condition2, BldgType, HouseStyle, RoofStyle, RoofMatl, Exterior1st, Exterior2nd, MasVnrType, Foundation, Heating, Electrical, GarageType, Fence, MiscFeature, SaleType, SaleCondition
Ordinal Qualitative	Street, Alley, LotShape, LandSlope, OverallQual, OverallCond, ExterQual, ExterCond, BsmtQual, BsmtExposure, BsmtFinType1, BsmtFinType2, HeatingQC, CentralAir, KitchenQual, Functional, FireplaceQu, GarageFinish, GarageQual, GarageCond, PavedDrive, PoolQC
Discrete Quantitative	MSSubClass, LotFrontage, LotArea, YearBuilt, YearRemodAdd, MasVnrArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, Bedroom, Kitchen, TotRmsAbvGrd, Fireplaces, GarageYrBlt, GarageCars, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, MiscVal, MoSold, YrSold

Some data is qualitative and some is quantitative, some data is numerical and other data is represented by strings, some data is ordinal, and other is nominal. In order to allow for a machine to easily process this data in a machine learning data, it must first be converted into data that is useful in an algebraic manner because that is the data that the machine learning algorithms we plan to use are made for.

To do this, the most important step is to first understand the data. There were many different ways in which similar data was presented in these variables, some examples of this are listed below.

- Binary Data - Different symbols mean the same thing
    - Street: Grvl(Gravel), Pave(Paved)
    - PavedDrive: N(Gravel), Y(Paved)
- Nominal Data - All nominal data used different types of naming systems
- Ordinal Data - Different ordinal data had different metrics
    - OverallQual: uses a 1-10 scale
    - ExterCond: uses Ex, Gd, TA, Fa, and Po as a ranking system
    - LandSlope: uses Gtl, Mod, Sev
- Discrete Quantitative -
    - GarageYrBlt - can have NA value that does not necessarily mean 0
    - GarageCars - can have NA value that means 0
    - YearBuilt - cannot have NA value

After fixing the non-nominal data so that the program can read it, the difficulty of dealing with the nominal data comes in. In order to deal with the nominal data, we used one hot encoding techniques to deal with this data.



	Name	Generation	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5	Gen 6
4	Octillery	Gen 2	0	1	0	0	0	0
5	Helioptile	Gen 6	0	0	0	0	0	1
6	Dialga	Gen 4	0	0	0	1	0	0
7	DeoxysDefense Forme	Gen 3	0	0	1	0	0	0
8	Rapidash	Gen 1	1	0	0	0	0	0
9	Swanna	Gen 5	0	0	0	0	1	0

As there were only 79 variables, all preprocessing was done manually using excel.

Selecting Different Machine Learning Algorithms

The preprocessing of the data indeed gave sharp direction to how the data would be approached when the regression algorithms would be applied to it. First, the project required that either python or R be used to process the data. Choosing python, the sklearn package was discovered, providing a number of machine learning functions. Originally, linear regression was chosen as the regression formula the would be run on the processed data; however, when it was discovered that the data was not of just numerical type, we realized different techniques used will provide different scores. This became a large part of our reality when we were attempting to remove outliers and looking at correlation between the data of a specific house and its sales prices. See Fig 1 as an example.

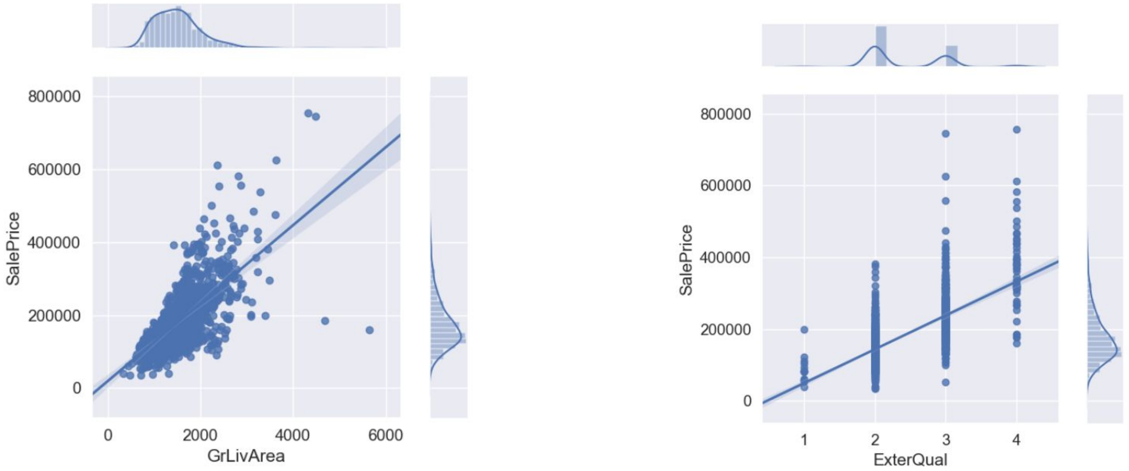
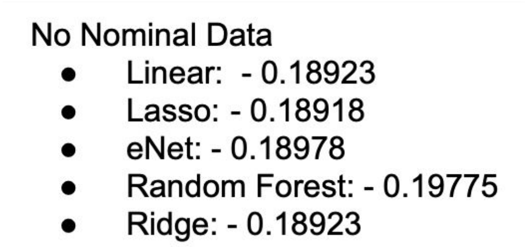


Fig 1:

It can see that the GrLivArea (Garage Living Area) in correlation to the sales price provides data that could be properly used with a linear regression model, yet the ExterQual (External Quality) in correlation to sales price would not be data that should be used with a model like linear regression. Initially unsure of how to approach the data that did not fit a linear model, we looked at the possibility of different regression techniques we could apply. Luckily, because of python we were able to approach the data with a number of different techniques using the sklearn package. Specifically, in addition to linear, we used random forest regression, elastic net regression, ridge regression, lasso regression, and gradient boost. With these different regression techniques provided by the sklearn package we looked into some possibilities of how they could benefit our data. In the beginning we thought that Random Forest and Gradient Boost would give us the best result. This was solely from our initial general knowledge. The fact that random forest or gradient boost may be able to handle non linear data in ways that other regression techniques would only be able to use linear data gave us high hopes for each technique. We also thought that the regression algorithms like Lasso and eNet would also give some result possibly better than linear, but similar because of the fact that a lot of the data was nonlinear.

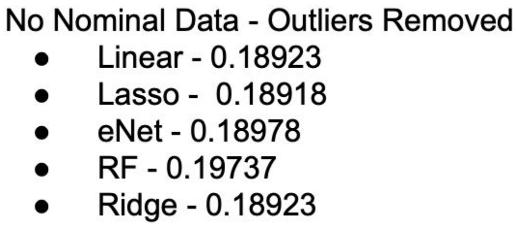
Results

The results held true to most of ideas and beliefs. Unsurprisingly, the gradient boost exceeded all expectations regarding the other algorithms. All models that fit a linear regression model reported almost the same score and the same result. The only big shock was that random forest provided a higher score than everything else.



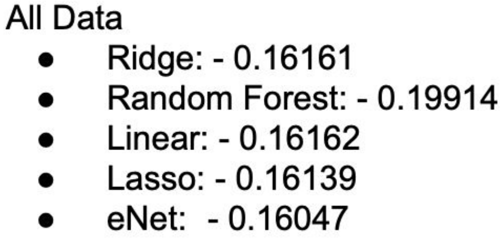
No Nominal Data

- Linear: - 0.18923
- Lasso: - 0.18918
- eNet: - 0.18978
- Random Forest: - 0.19775
- Ridge: - 0.18923



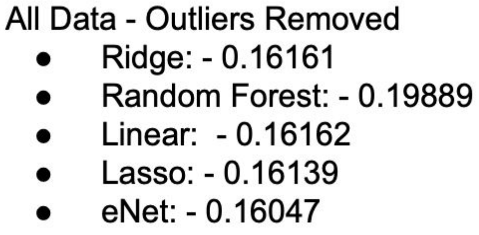
No Nominal Data - Outliers Removed

- Linear - 0.18923
- Lasso - 0.18918
- eNet - 0.18978
- RF - 0.19737
- Ridge - 0.18923



All Data

- Ridge: - 0.16161
- Random Forest: - 0.19914
- Linear: - 0.16162
- Lasso: - 0.16139
- eNet: - 0.16047



All Data - Outliers Removed

- Ridge: - 0.16161
- Random Forest: - 0.19889
- Linear: - 0.16162
- Lasso: - 0.16139
- eNet: - 0.16047

What We Learned:

A lot was learned in the process as we faced many obstacles and difficulties. From the large quantity of data being used and its many differing types, we learned how to correctly preprocess it into easily useable chunks. Our experience with handling and analyzing data for proper regression was minimal prior to the project, and so this was a struggle at first. But, due to the struggle our learning was enhanced further. Python experience prior to this project was minimal, and the only previous experience with coding regression algorithms was from the previous homework given in the class. So, going over that to aid us in this project was helpful to our learning process and useful in the project. Overall, we learned a lot about machine learning, we hope to continue to use this knowledge after this class has ended, and we also wish to further increase our knowledge of machine learning in the future.