HIGH SCHOOL EDITION
Journal of Student Research
Volume 11 Issue 4 (2022)

Using Supervised Machine Learning to Predict House Prices

Alexander Tsai[1] and Hieu Nguyen[#]

[1] Hunter College High School, New York, NY, USA
[#] Mentor

ABSTRACT

Given the recent influx of prices in the housing market, determining a fair housing price has been of high interest for many homebuyers and sellers alike. In this project, various machine learning models are used to predict the price of a house based on physical features and characteristics such as lot size and neighborhood. Extensive data preprocessing and feature engineering were employed to aid the models' performance compared to other models in the market. The best models have been able to predict U.S houses' prices within a RMSE value of $23,000 when the mean price of a house in the dataset is $180,000. In future research, this model can be implemented in various other places within the U.S and additional features can improve performance further.

Introduction

Ever since the housing market crash of 2008, house pricing has been an important point of focus for homebuyers and economists alike. The increasing difficulty to buy homes makes it ever more important to ensure a buyer is getting a fair price for a property. Artificial intelligence and machine learning have also taken a forefront in the public eye and are nearly everywhere in everyday life now. From social media algorithms to google search autocomplete and spell check, artificial intelligence is becoming an integral part of human life. Classification and regression are two other common uses of machine learning, and teaching a machine to do tasks like these can save lots of time.

In this project, supervised machine learning models are used to predict the housing prices of homes in Ames, Iowa. The regression models linear regression, lasso, ridge, elastic net, random forest and support vector machines are all used with varying effectivenesses to predict sale prices based on a number of qualitative and quantitative physical features of the property. Furthermore, data preprocessing, feature selection and feature engineering are all utilized to modify the dataset to increase the learning efficiency of the machine learning models. After manipulating the dataset and optimizing the models, a significant improvement in prediction accuracy was noticed across many models. The paper is structured as follows: section 2 is about data processing and feature engineering, section 3 describes the methods and models used, section 4 analyzes the results, and the conclusion is in section 5.

Data

Data Exploration

The first step is to visualize the distribution of our dataset, which is a collection of houses in Ames, Iowa. The dataset was obtained through Kaggle [1]. The histogram and data table indicate that 50% of houses are within

the 130,000 to 220,000 dollar range with a mean of 180,000. This is important because it indicates the benchmark of which to compare the results to: a smaller range of prices calls for stricter rates of error. Furthermore, the maximum price of a house in this dataset is nearly 755,000, roughly 4x the mean. Since the range of prices from the 75th percentile to 100th percentile is massive, removing outliers may prove beneficial in training the model. Houses whose cost was 3 standard deviations from the mean were removed from the dataset. In other words, the bottom and top 0.1% of houses were removed from the dataset. After this, the 1460 data points in the original dataset was reduced to 1438, removing 22 outliers.

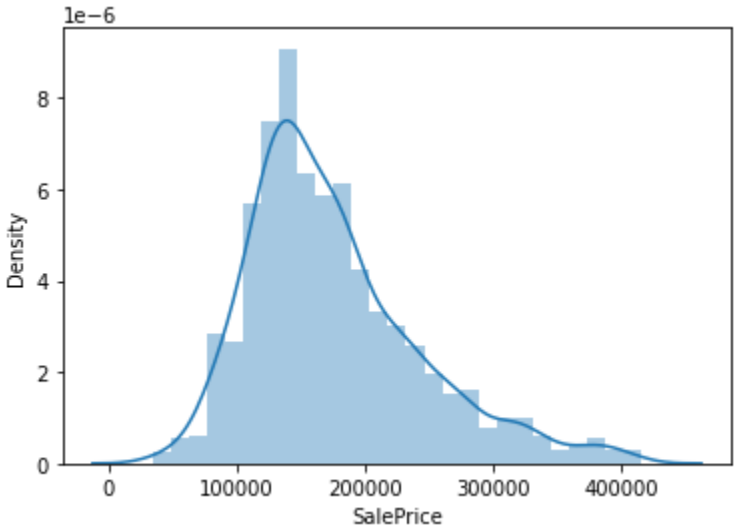

Figure 1. Sale Price distribution. Most of the house prices fall within $100,000 to $300,000 with the mean being about $180,000.

Analyzing the distribution of data also enables contextualization of results. The margin for error is directly proportional to the magnitude of the data. As the scope of the data is in the 100,000s range, an acceptable margin for error would be about 1 magnitude lower, perhaps in the 10,000s. That being said, a model with an even lower margin of error may not actually be a better model. A common issue in machine learning is that of overfitting, where the model is overtrained and performs extremely well with one set of data, but extremely poorly with other sets because the weights of the model have been tuned too closely to a specific set of data.

Table 1. An overview of the original dataset before any preprocessing .

Statistics	**Values**
Count (Datapoints)	1460
Mean ($)	180,921
Standard Deviation ($)	79,442
Minimum ($)	34,900
25th Percentile ($)	129,975
50th Percentile ($)	163,000
75th Percentile ($)	214,000

Maximum ($)	755,000

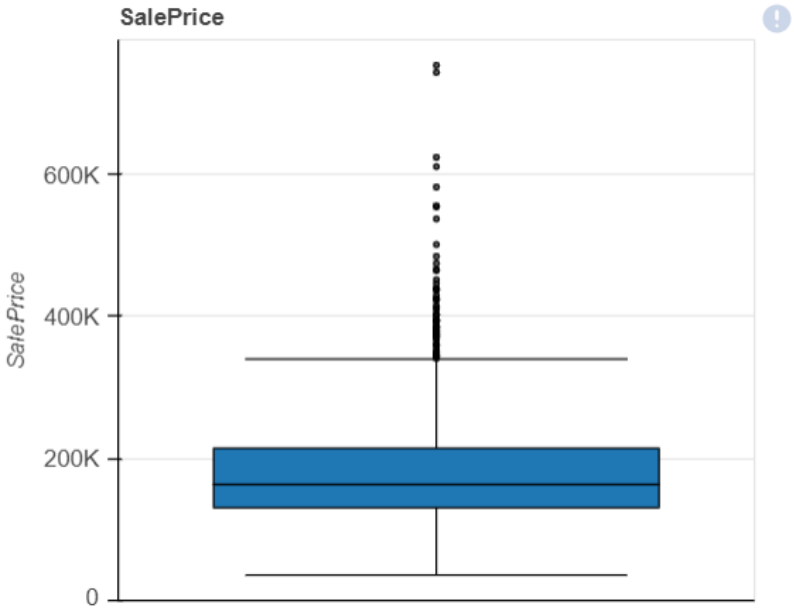

Figure 2. Sale Price boxplot. Using the visualization of the boxplot, we are able to identify outliers and exclude them from the final training dataset.

Data Preprocessing

While the type of learning that different models use is a large factor in its trainability, the digestibility of the dataset is equally important. The analogy of teacher-student interaction can help visualize this relationship: the teacher's overall knowledgeability limits how much the student can learn, how well the teacher can explain ideas to the student limits how quickly the student can learn. Both are incredibly important to the final result. There are an abundance of methods and strategies that can be employed to help the model understand the data better. The ones that were employed in this project are variable selection and feature engineering.

While all 80 of the house characteristics play a part in determining its price, for the purposes of training a model to predict prices, it is a lot more efficient to focus on the factors that play a more significant role in determining the final price. This can be accomplished by running a comparison between how each characteristic relates to sale price, also known as the correlation between each characteristic and sale price.

The below figure (Fig.3) describes how closely correlated each qualitative characteristic is with each other. On each axis, there is a list of all the variables and the corresponding intersections between them represent the correlation value. There are two types of correlation on this chart: linear correlation and inverse correlation. Linear correlation, represented by blue, is when both variables increase with one another and inverse correlation, which is represented by red, is when one variable decreases as another increases. The Pearson Correlation Coefficient, which is a value between -1 and 1 that is represented by the intensity of the color on the chart. The "Sale Price" row and column are at the bottom and rightmost, and from figure 3, the factors that are the most influential in determining price can be determined. The threshold used as the cutoff was an absolute value of 0.1.

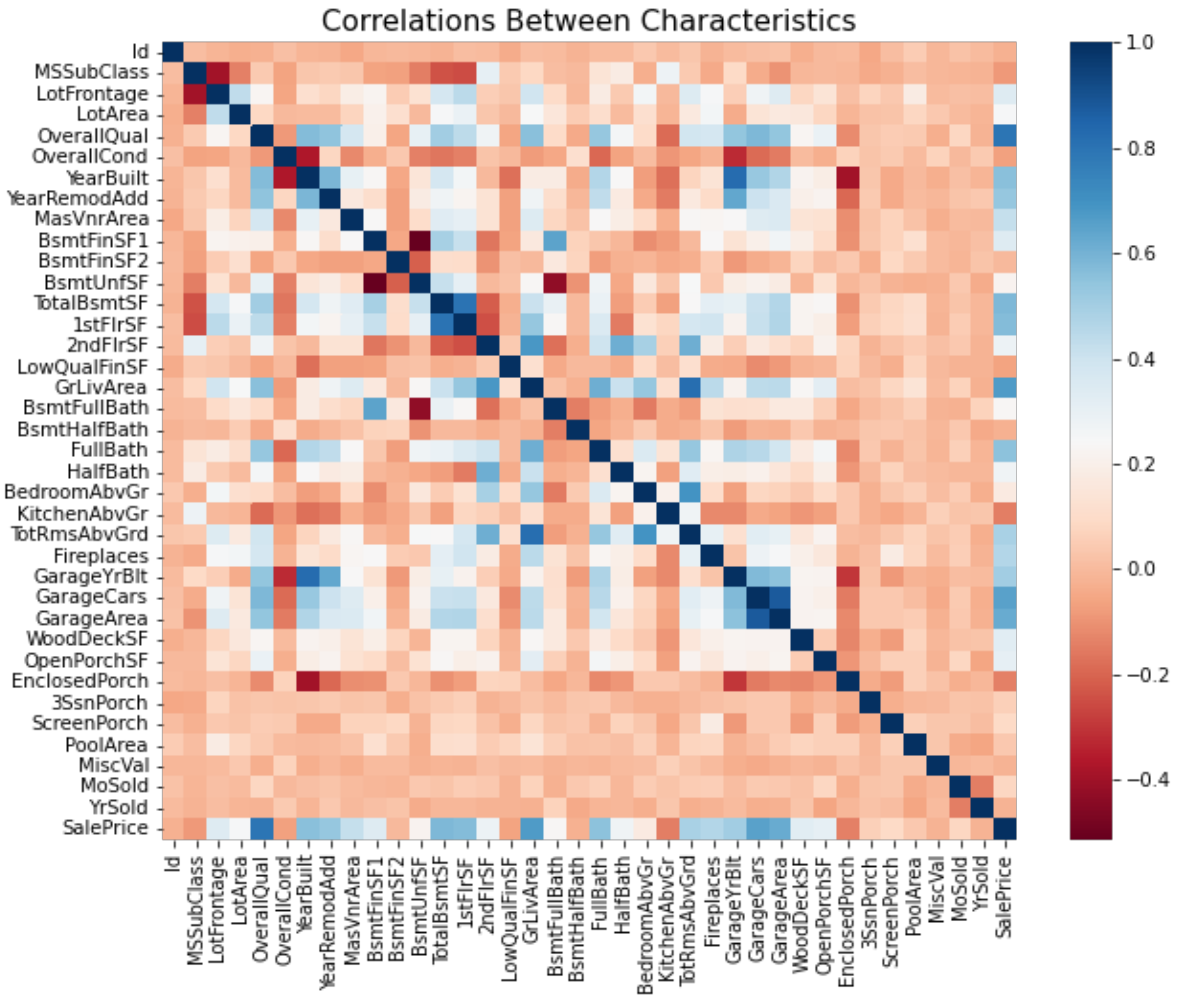

Figure 3. Initial Correlation Matrix. Using the correlation matrix, we can identify which characteristics are important, and then get rid of the unimportant ones.

Feature Selection

Unfortunately, numerical correlation can only be applied when the data is, as its name suggests, numerical. In the dataset, there are many categorical variables that play a significant role in helping predict house price, for instance, houses in the same neighborhood tend to cost similar prices. Categorical variables were looked at by hand, and were analyzed based on intuition. The categorical variables that seemed important were the following: "MSZoning", "Utilities", "BldgType", "Heating", "KitchenQual", "SaleCondition", "LandSlope", "Neighbor-Hood"

After excluding uncorrelated characteristics, the final important features are reduced to the following:

Table 2. Important Features.

Feature	Description	Feature	Description

Lot Area	Area in front of the property	Fireplaces	Number of fireplaces
OverallQual	General condition of the property	WoodDeckSF	Area of wood deck
YearBuilt	Property's age of construction	OpenPorchSF	Area of open porch
YearRemodAdd	Year of most recent property renovation	EnclosedPorch	Area of enclosed porch
BsmtFinSF1	Finished square feet in the basement	GarageAreaCar	Area of garage times area of cars
BsmtUnfSF	Unfinished square feet in the basement	GarageArea	Area of the Garage
TotalBsmtSF	Total square feet in the basement	GarageCars	Number of Cars that can fit in the garage
1stFlrSF	Area of the first floor	MSZoning	Type of area property is located in
2ndFlrSF	Area of the second floor (if applicable)	Utilities	Available utilities
GrLivArea	Total area that is above ground level	BldgType	Type of building
BsmtFullBath	Full bathrooms in the basement	Heating	Type of heating system
FullBath	Full bathrooms above the ground	KitchenQual	Overall quality of the kitchen
HalfBath	Half bathrooms above the ground (No shower)	SaleCondition	Type of transaction
BedroomAbvGr	Number of bedrooms above ground	LandSlope	Slope of the property

KitchenAbvGr	Number of kitchens above ground	Neighborhood	Name of the neighborhood
TotRmsAbvGrd	Total number of rooms above ground		

Feature Engineering

It is important to avoid overrepresenting variables in the final dataset. Characteristics like “GarageCars” and “GarageArea” are going to be closely related because the number of cars in a garage is limited by the size of the garage. As seen in the correlation chart, both of these characteristics are pretty closely related to sale price. Even if both of those characteristics heavily influence sale price, it doesn’t make sense to include both of them in our final data. This is because it increases the difficulty of training the model because the importance of the garage is overrepresented when the garage is accounted for twice. This can counter this in two different ways: removing one of the two or factor engineering. The approach taken in this project was the latter and the two variables into a singular variable. A better indicator of garage value might be the ratio of “GarageCars” to “GarageArea”, a value that yields the amount of area each car in the garage receives.

The last step to preprocessing the data is to remove variables that are missing among numerous data points. These missing data points can make training the model extremely challenging and can introduce lots of bias into the system.

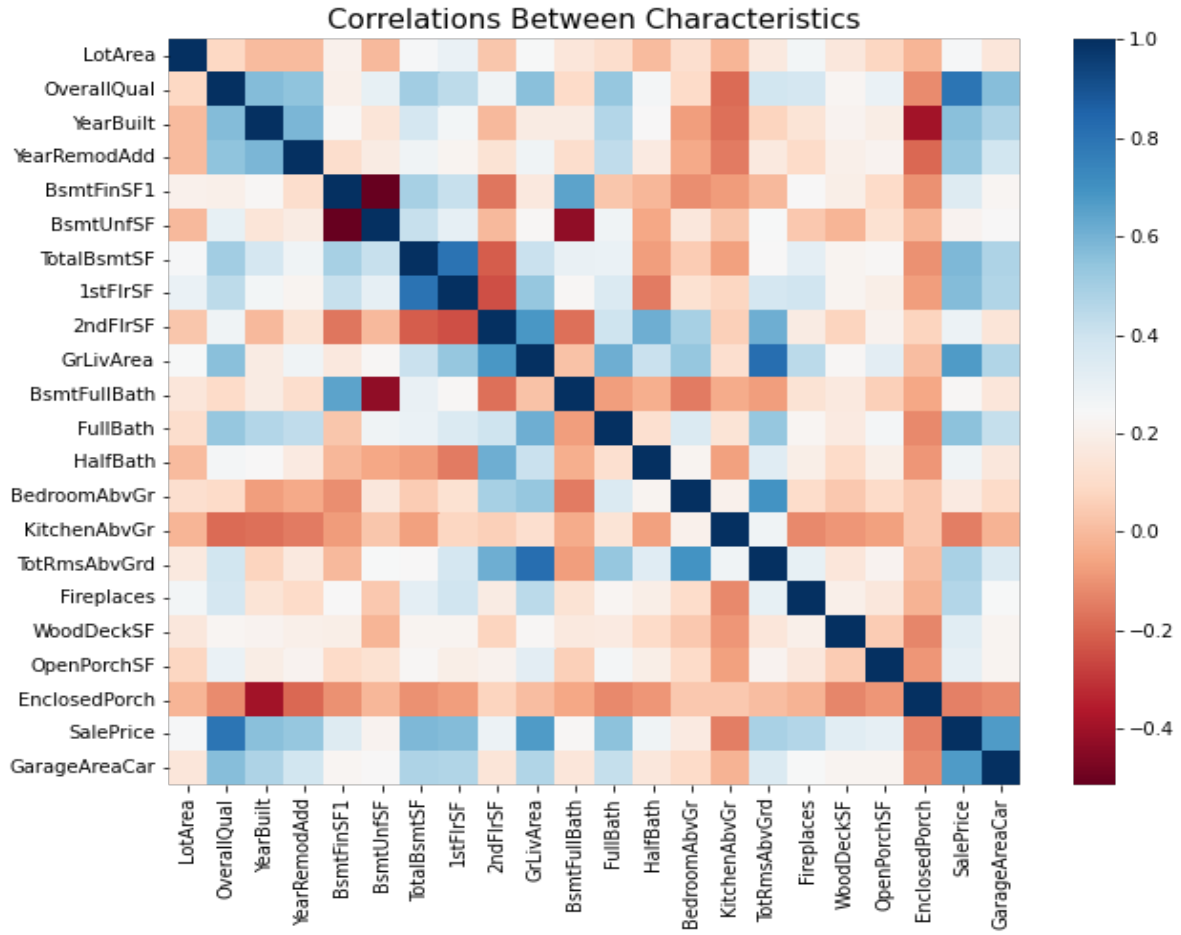

Figure 4. Updated Correlation Matrix. After eliminating variables with low absolute correlation to "SalePrice", 21 quantitative variables are left for use in training.

Methods

Now that the data has been properly preprocessed, different machine learning models can finally be employed to actually predict prices. A variety of different learning models were used to predict the sale price to observe which models perform the best for this project. The python library scikit learn was used and provided the following models [2].

Linear Regression: (LR) aims to find the curve that best fits the data. This is done by determining the coefficients and variables such that the relationship between the dependent and independent variables is optimally described. The model estimates the best curve by minimizing the residual sum of squares between the labels provided by the data and the targets predicted by the linear approximation.

Lasso: Lasso regression, also known as penalized regression, is a version of linear regression that attempts to lower the total value of the weights, commonly lowering the weights of many variables to 0, removing features and also acting as a pseudo-feature selection. Lasso generally works better in a model with less features than observations.[3]

Ridge: Ridge regression is very similar to Lasso regression, except that it takes the squared value of the weights instead of just the value. Ridge will also try to lower weights, but will not reduce them to 0. Ridge generally works better in a model with more features than observation [4]

Elastic Net: Elastic net takes features from both Lasso and Ridge, and tries to find a middle ground between the two models. Elastic net tries to utilize the strengths of Lasso and Ridge while mitigating their weaknesses [5]

Random Forest Regression: Random forest is a forest of decision trees that averages each individual tree's prediction. It is used for both classification and regression [6]

Support Vector Regression: Support Vector Regression (SVR) is a non-parametric regression that relies on kernels. The line of best fit drawn in in SVR is the one where a hyperplane crosses the most data points[7]

Each model was trained under two datasets, one with and one without feature engineering and preprocessing. Each model was then cross-validated with data that was not in the training data set and the Root Mean Square Error (RMSE) was recorded. In this context, the RMSE value denotes the difference between the value predicted by the model and the actual price of the house. RMSE favors consistent models with lower variances. As opposed to Mean Average Error (MAE), RMSE punishes differences that are larger more heavily than smaller differences

Results

Table 2. Important Features.

Model	RMSE (before feature engineering & data preprocessing)	RMSE (after feature engineering & data preprocessing)
LR	33,335	31,616
Ridge	33,101	24,025
Lasso	33,206	24,069
Elastic Net	39,383	27,300
SVR	29,342	24,482
Random Forest	28,817	27,300

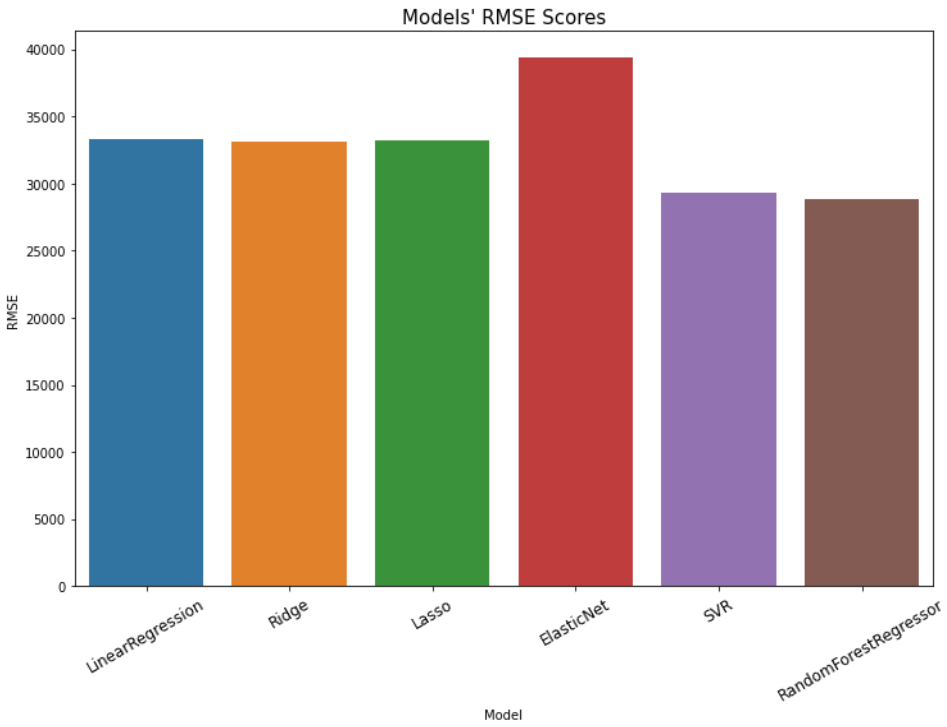

Figure 5. Initial RMSE Bar Graph.

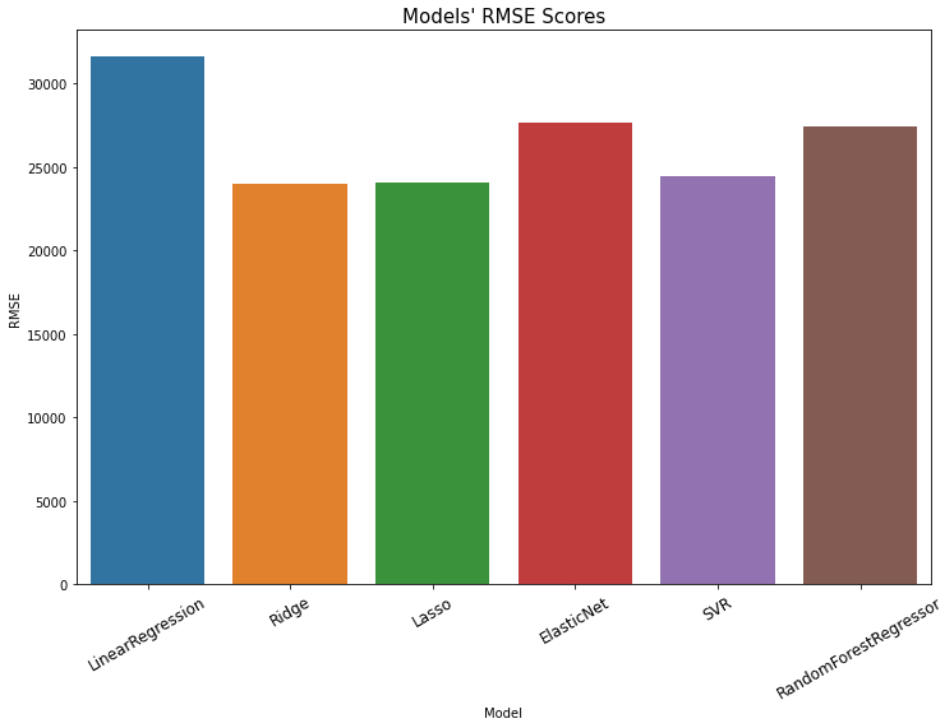

Figure 6. Updated RMSE Bar Graph.

As seen in the table above, there was a general increase in performance after feature engineering and data preprocessing with varying amounts of improvement across different models. Before feature engineering and preprocessing, the best performing models were the non-parametric models SVR and Random Forest, each scoring a RMSE value of around 29,000. Linear Regression, Ridge and Lasso all performed very similarly to one another, as the latter 2 are a regularized version of the first. It is also interesting to note how Elastic Net, a compromise between Lasso and Ridge, performed worse than both Ridge and Lasso. This relationship carried on even after feature engineering and preprocessing.

After feature engineering and preprocessing, Ridge, Lasso and SVR performed the best, with each scoring a RMSE value of about 24,000.. Elastic Net noticed the most improvement whereas Random Forest and Linear Regression noticed little change. The difference in improvement between Linear Regression and Ridge and Lasso is astonishing, and demonstrates the importance of regularization in a preprocessed dataset.

A MDI test was also run using the random forest model to isolate the individual importances for each feature. From the bar graph in Figure 7, we can see that the "LotArea" was the most important feature in determining house price, followed by "OverallQual."

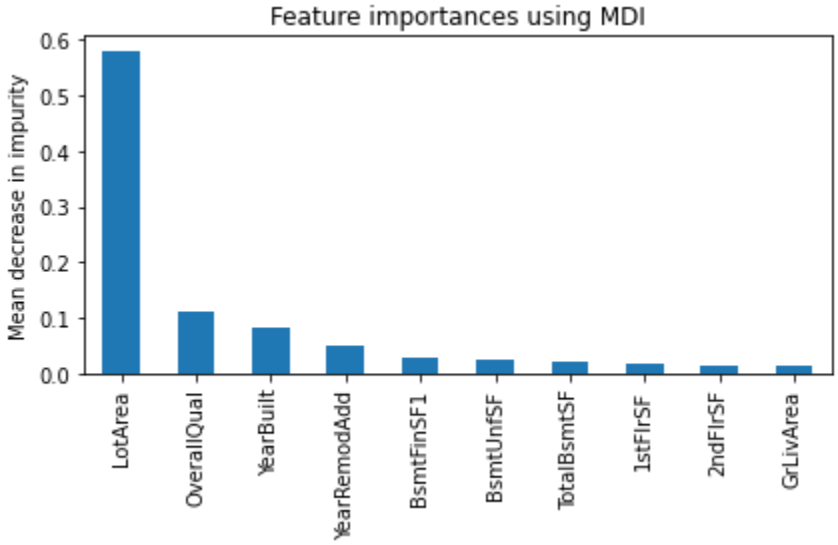

Figure 7. Feature Importances Bar Graph. By graphing the MDI values each feature, we can identify which features play the largest role in determining the model's prediction

Conclusion

After data preprocessing and feature engineering, we were able to train different machine learning models to predict housing prices within 13% on average. Compared to the initial prediction accuracy of 20%, data preprocessing and feature engineering proved to play a significant improvement in training machine learning models. While the models can give a reasonable estimate as to the price of the house, using regression cannot fully encapsulate the scenario the house is seated in. The price of the house is usually derived from a variety of factors including the physical features of the house but does not really give the full context: housing in cities or suburban areas are generally more expensive than those in rural areas. The dataset was heavily biased towards more suburban housing and the consequent models produced may not work as well with urban housing. Furthermore, the housing market is constantly changing. The dataset used in this project may become

quickly outdated. That being said, the overall concepts remain unchanged, and through data preprocessing and feature engineering, supervised machine learning models can be trained to accurately predict housing prices.

Acknowledgments

I would like to thank my mentor Hieu Nguyen for teaching me the fundamental concepts behind supervised machine learning and inspiring me to write this paper.

References

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

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830

[3] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12*(1), 55-67. https://doi.org/10.1080/00401706.1970.10488634.

[4] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[5] Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301-320. https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[6] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689-702. https://doi.org/10.1016/j.ejor.2016.10.031

[7] Huang, W., Lai, K. K., Nakamori, Y., & Wang, S. (2004). Forecasting foreign exchange rates with artificial neural networks: A review. *International Journal of Information Technology & Decision Making*, *3*(01), 145-165. https://doi.org/10.1142/S0219622004000969