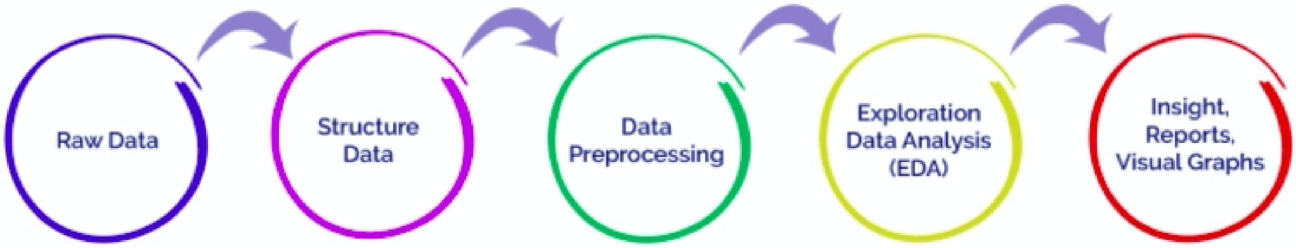
Manarat International University
Department of Computer Science and Engineering
Artificial Intelligence (CSE – 411)
Project report

Problem Tile:	House Prices: Advanced Regression Techniques
Name of the Team:	**rdnasim**
Contestants Name:	Riadul Islam Nasim Mahfuzur Rahman
Student ID:	1640CSE00467 1640CSE00519
Kaggle Account:	rdnasim
Git Repository link:	rdnasim

1 Project Goal

The goal for the competition is to predict the sales price for each house. For each Id in the test set, must predict the value of the SalePrice variable.


Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

1.1 Problem statement

The problem will be addressed if we build predictive models using Advanced Regression Techniques and train the model and pick the best model using validators so that it can accurately predict the value of House price. The data set is obtained from Kaggle and it is a Competition Data set which contains 79 features which influences the price of Homes at Ames, Iowa. The competition organizer also boasts about the advantages of not just estimating house price using number of bedrooms or the fence around the house usually done by the brokers. Yeah, he is right, when you can accurately predict the house price using Advanced Regression techniques then why bother about a House Broker's price estimate?

I started this competition by just focusing on getting a good understanding of the dataset. The EDA is detailed and many visualizations are included. Most of the work done dealt with transforming, cleaning, imputing, and aggregating data. Once complete, models were made using the following machine learning algorithms. This version also includes modeling:

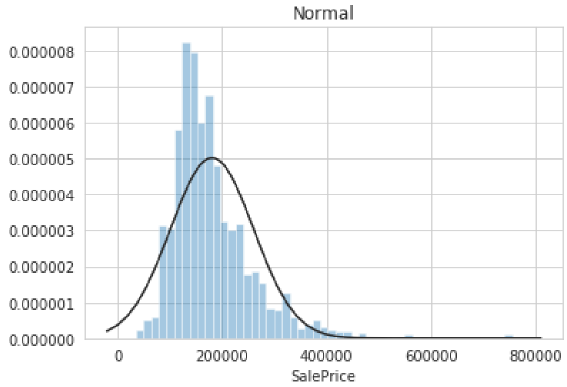
- START Fit
- stack_gen
- elasticnet

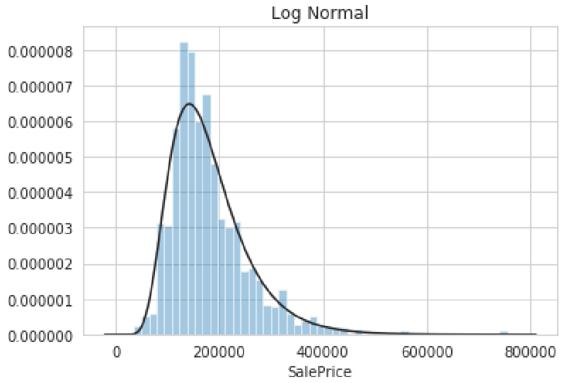
- Lasso
- Ridge
- Svr
- GradientBoosting
- xgboosta
- lightgbm

2 Data Preprocessing

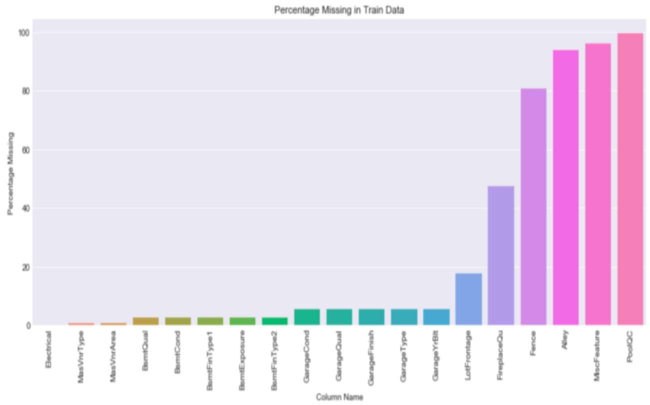
Dataset	Number of features	Number of observation
Train	81	1460
Test	80	1459

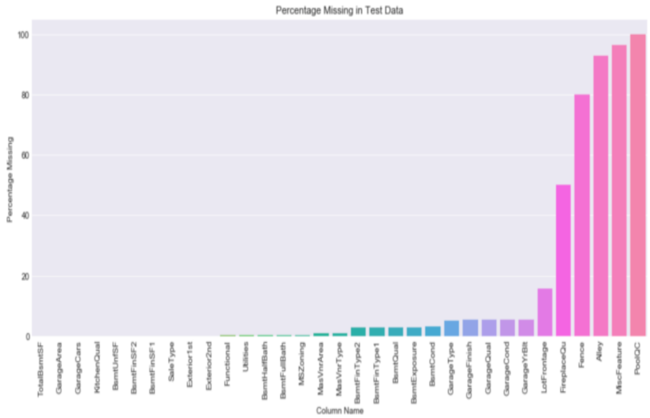
A histogram plot shows the distribution of the target variable ‘SalePrice’ as being was right-skewed. Before moving any further, I decided to obtain a normal distribution by way of log-transformation.





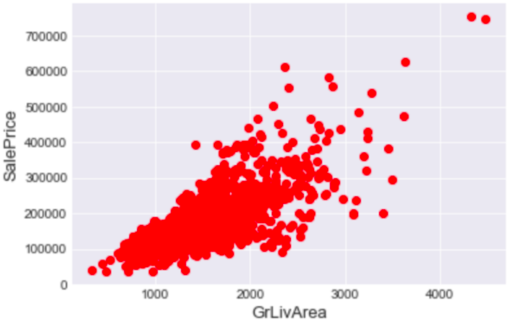
To get a better understanding of missingness with the given data sets (train and test), I determined the percentage of missing values and built bar plots.





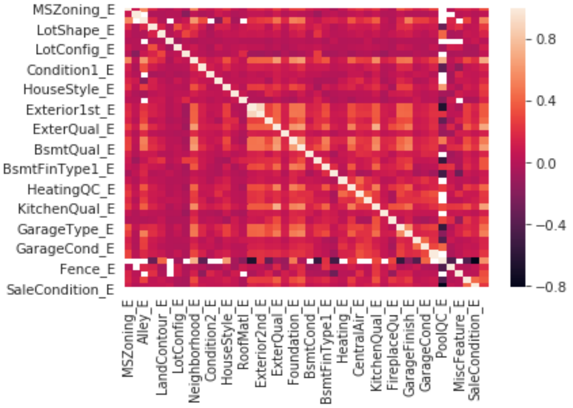
In the train dataset, 19 out of 81 features had missing values. As for the test dataset, 33 out of 80 features had missing values.

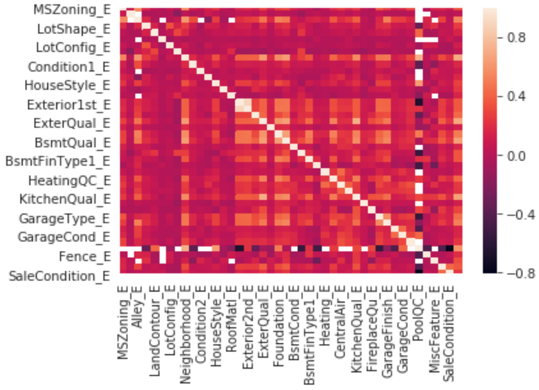
Some of the features had outliers. Looking at the graph below we can see that there are two homes that are very spacious yet are extremely low in price. When making my model I must ensure that they are robust to outliers.

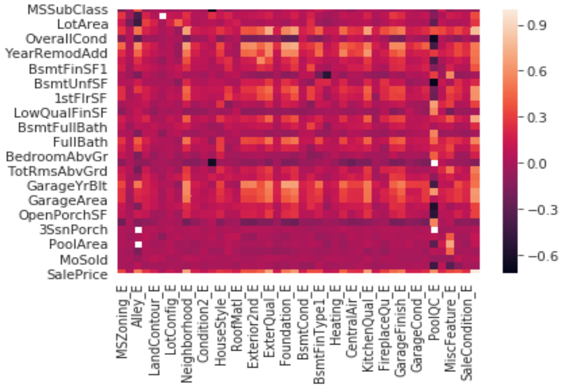


To check for multicollinearity, I created a correlation matrix with respect to the target variable ‘SalePrice’.

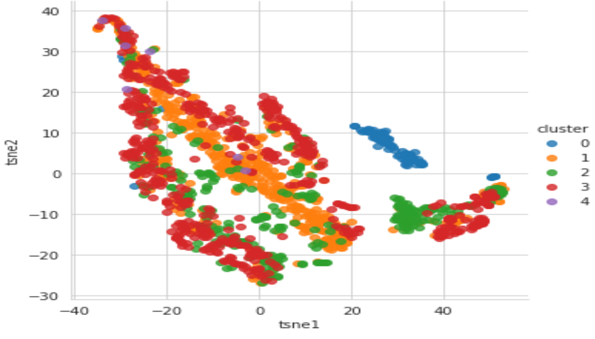
Some variables were highly correlated such as 'GarageArea' and 'GarageCars'. This makes sense since the size of a garage determines('GarageArea') the number of cars that fit in it ('GarageCars'). Other variables that were highly correlated show the same type of dependency.







Simple clustering - Explained_variance_ratio is 0.754827252847



Missingness

As you can see the categorical features ‘Alley, ‘PoolQC’, and ‘MscFeature’ have over 90% of its column missing. However, looking at the data description, the NA’s in categorical variables actually mean “not present”. For example, NA in the ‘Alley’ variable meant that the home has no alley.

To correct the issue, I first concatenated the train and test set. I then replaced the Na’s with “None”. Researching further, I realized that two of the numerical features (‘MSSubClass’, ‘Mosold’) were actually categorical. ‘MoSold’ used numerical values represented months. For example, “1” meant January and “2” meant February. In ‘MSSubClass’, numerical values identified the type of dwelling involved in the sale. Both features were converted into strings. For the remaining numerical features, NA’s were replaced by the value zero.

3 Feature Engineering

Since the data has many categorical variables, I converted them into dummy/indicator variables. As for numerical features, quite a few had a highly skewed distribution which can lead to poor models. To combat this, these features were log transformed. Removing features that are not very useful . This can be understood only by doing proper EDA on data. Adding new features for the better performance. Concatenation the train and test data in the same dataframe, Create features, Check how the features , work with the model and Log transformation.

4 Modeling Methods

Time to build some models! We began by creating some benchmarks using a Linear Regression model on both the scaled and non-scaled data. We then prepared a series of fits using many regularized linear regression models. The models we fit were:

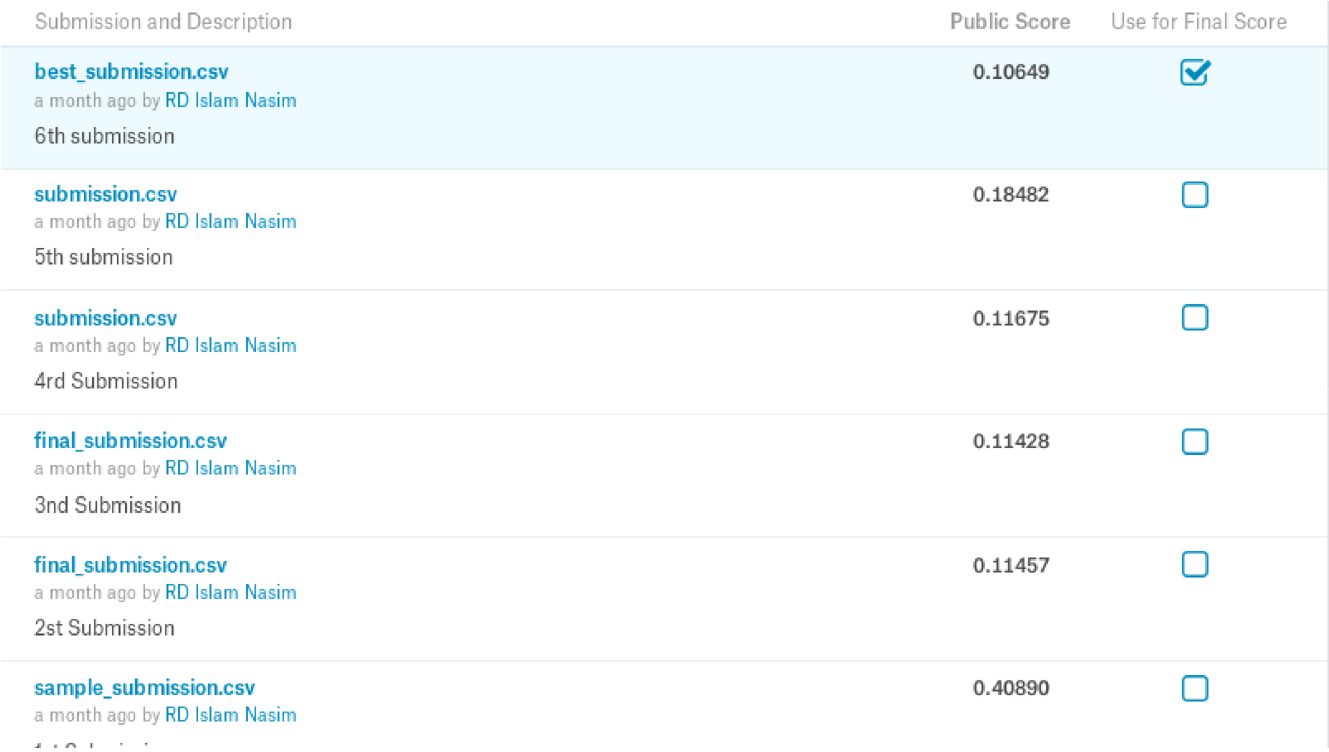
- A naive Ridge Regression
- A naive Lasso Regression
- A naive ElasticNet Regression
- A naive Ridge Regression
- A naive Lasso Regression
- A naive ElasticNet Regression
- A naive Svr Regression
- A naive GradientBoosting Regression
- A naive xgboost Regression
- A naive lightgbm Regression
- A naive stack_gen Regression

5 Results & Discussion

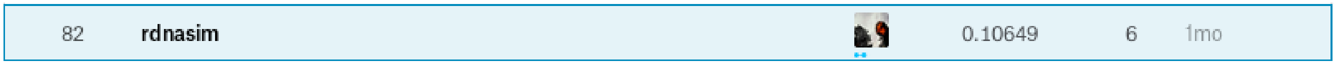
Our top performing models were:

Dataset	Model	Preprocessing	Score
test	Kernel Ridge	scaled	0.1024
test	Lasso	scaled	0.1147
test	ElasticNet	scaled	0.1031
test	SVR	scaled	0.1033
test	Lightgbm	scaled	0.1066
test	GradiantBoosting	scaled	0.1072
test	Xgboost	scaled	0.1064

“Top performing models. Score is the R2 score of each test”



Submission and Description	Public Score	Use for Final Score
best_submission.csv a month ago by RD Islam Nasim 6th submission	0.10649	☑
submission.csv a month ago by RD Islam Nasim 5th submission	0.18482	□
submission.csv a month ago by RD Islam Nasim 4rd Submission	0.11675	□
final_submission.csv a month ago by RD Islam Nasim 3nd Submission	0.11428	□
final_submission.csv a month ago by RD Islam Nasim 2st Submission	0.11457	□
sample_submission.csv a month ago by RD Islam Nasim	0.40890	□



82	rdnasim	0.10649	6	1mo