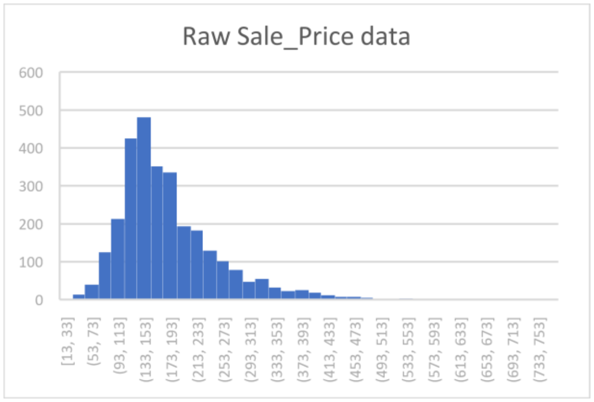
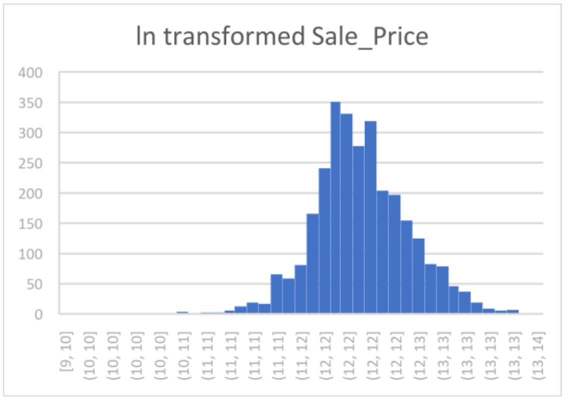
AMES Housing Dataset Modeling

This version of the AMES Housing dataset is downloaded from the CS 598 (Stat 542) Piazza forum under the resources tab. It contains 2,930 records of housing sales and 83 explanatory variables that describe many characteristics of houses. The `Sale_Price` column will be the response variable for this report. The `Sale_Price` in the dataset can be seen to exhibit some skew and can not be modeled as a gaussian distribution, so this report applies a log transformation to attempt to make the response more of a normal distribution as can be seen in the below histograms. The purpose of this project is to accurately predict the `Sale_Price` of a house (in log scale.)





Pre-processing of data

The data provided is a relatively clean datatset. Only one column can be seen to have missing values – `Garage_Yr_Blt`. The missing values were replaced with 0's. The justification for this is that these values correspond to `Garage_Finish`, `Garage_Qual`, and `Garage_Cond` columns having the value of 'No_Garage', so that any correlation of this variable on `Sale_Price` can be accurately modeled by these other features.

Many of the features are strings representing the characteristic of a particular feature. These variables cannot be treated as numerical entries for modelling purposes. This report converts these categorical features into multiple columns representing the presence (1) or absence (0) of each value these features can take. All features found to have character values are converted in this fashion.

The below 46 categorical features were split to make enough columns to represent all of the different values that these features can take. If the feature only has two values, then only one column was added

MS_SubClass	MS_Zoning	Street	Alley	Lot_Shape	Land_Contour	Utilities	Land_Slope
Neighborhood	Condition_1	Condition_2	Bldg_Type	House_Style	Overall_Qual	Overall_Cond	Roof_Style
Roof_Matl	Exterior_1st	Exterior_2nd	Mas_Vnr_Type	Exter_Qual	Exter_Cond	Foundation	Bsmt_Qual
Bsmt_Cond	Bsmt_Exposure	BsmtFin_Type_1	BsmtFin_Type_2	Heating	Heating_QC	Central_Air	Electrical
Kitchen_Qual	Functional	Fireplace_Qu	Garage_Type	Garage_Finish	Garage_Qual	Garage_Cond	Paved_Drive
Pool_QC	Fence	Misc_Feature	Sale_Type	Sale_Condition	Lot_Config		

As an example, the `Lot_Shape` feature can take on the following values: `Irregular`, `Regular`, or `Slightly Iregular`. Three separate columns were made for these values.

For Tree based modelling, cleaning up the missing values and adding the categorical values columns, was the only pre-processing that was done. Random Forest methods proved challenging to meet the target thresholds. Boosting tree methods proved to be acceptable in meeting the target error rates. In particular, *xgboost*, from the **xgboost** package is one of the models used in this final report. xgb.cv was used to get the number of rounds to use for the model. The final xgb model uses the following parameters:

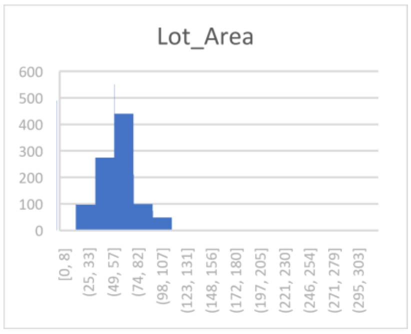
eta=0.05 – the learning rate, which controls the shrinkage rate of the parameters at each step
gamma=0 – controls the regularization, here there is no regularization penalty
max_depth=6 – the maximum number of levels in each tree
subsample=0.5 – the number of samples within the data to supply to each tree, here 50%
nrounds=700 – the number of boosting iterations required for convergence

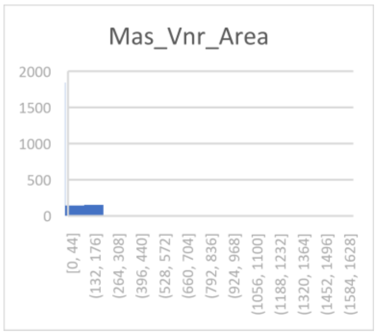
The error rates for the different train/test splits are documented at the end of this report.

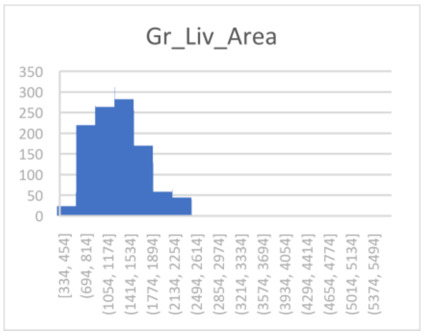
Linear models rely on the mean and variance of different features. Extreme values, can therefore have a significant impact on modelling accuracy. Winsorizing is a strategy to set all extreme values to a specified percentile of the data to limit the impact to the model accuracy. For this project, the 95th percentile was used as the limit. Any data points beyond the 95th percentile were set equal to the 95th percentile value. The following features were Winsorized in this project :

Lot_Frontage	Lot_Area	Mas_Vnr_Area	BsmtFin_SF_2	Bsmt_Unf_SF	Total_Bsmt_SF	Second_Flr_SF	'First_Flr_SF'
Gr_Liv_Area	Garage_Area	Wood_Deck_SF	Open_Porch_SF	Enclosed_Porch	Three_season_porch	Screen_Porch	Misc_Val

These features were chosen for winsorization based on the distribution of the data. Below are a few histograms to show that there are a few extreme values that could potentially effect the accuracy of the models.







Several attempts at models were tried with just filling in missing values, adding columns for categorical variables, and winsorization. After failing to meet the target thresholds for error rates, some features were considered for removal. Below is a summary of the variables that were removed and the rationale.

Feature	Reason For Removing
Longitude ***Latitude***	These variables can be captured in the Neighborhood feature
Roof_Matl ***Heating*** ***Pool_QC*** ***Misc_Feature*** ***Low_Qual_Fin_SF*** ***Pool_Area*** ***Street***	All of these variables had one state that dominated the dataset. Of the different categories in each of the features, less than 14 observations were attributed to another category than the main one. Their inclusion in any of the 10 train/test splits is likely to have very little impact on the model
Condition_2	This is almost identical to *Condition_1*

After several attempts, the **glmnet** package was used with some success in meeting the target error thresholds. The `cv.glmnet` function was used with the following parameters:

alpha = 0.2 - The elasticnet mixing parameter
lambda = (0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1) – The lambda sequence used to determine best lambda.

The error rates for the different train/splits are documented below.

Results

The below table shows the error rates for the different train/split sets for each of the final two models.

Model/Set	1	2	3	4	5	6	7	8	9	10
xgboost	**0.112**	**0.115**	**0.116**	**0.118**	**0.108**	**0.124**	**0.131**	**0.126**	**0.131**	**0.121**
cv.glmnet	**0.123**	**0.118**	**0.120**	**0.120**	**0.112**	**0.133**	**0.126**	**0.119**	**0.129**	**0.123**

The models were run under the following environment:

Computer Component	Component Characteristic
Processor	**Intel® Core™ I3-3227U CPU @ 1.90 GHz**
Installed RAM	**8.00 GB (7.89 GB usable)**
System type	**64-bit operating system, x-64-based processor**
Operating System	**Windows 10 Home, Version 2004, Build 19041.508**
Running time for 1 iteration	**Approx 28 seconds for both train and test data combined**

Observations

While the *xgboost* model takes more time to train than the *glmnet* model, overall it provides better prediction accuracy for the first 5 train/test splits. However, for the remaining 5 train/test splits they have very similar acuracy numbers.