House Price Prediction – Ames Iowa Dataset

CS580L-01 Fall 2019 Project Final Report

Siddhesh Kolhapure, Shweta Mestry, Aditya Sawant
[skolhap1, smestry1, asawant2]@binghamton.edu

1. Introduction

In today's world, buying a new house sounds like daunting task to us. Housing prices are an important reflection of the economy, and housing prices ranges are of great interest of both buyer and seller. There are various factors which decide the value of the property. In this project, house prices will be predicted based upon given 79 explanatory variables that covers many aspects of residential houses. Our project goal is to create models that can accurately estimate the price of the house with given features. We will be predicting Sales price of housing in Ames, Iowa using regression model like XGBoost "Extreme Gradient Boosting" and Artificial Neural Network "ANN (2 Hidden Layers, No Dropouts, No Batch Normalization Layers)".

2. Data and Preprocessing

The dataset is the prices and features of residential houses sold from 2006 to 2010 in Ames, Iowa, obtained from the Ames Assessor's Office. This dataset consists of 79 house features and 1460 houses with sale prices. Although the dataset is relatively small with only 1460 examples, it contains 79 features such as areas of the houses, types of the floors, and numbers of bathrooms.
Such large amounts of features enable us to explore various techniques to predict the house prices.

The dataset consists of features in various formats. It has numerical data such as prices and numbers of bathrooms/bedrooms/living rooms, as well as categorical features such as zone classifications for sale, which can be 'Agricultural', 'Residential High Density', 'Residential Low Density', 'Residential Low Density Park', etc. In order to make this data usable with the model, categories of the categorical data are converted in numerical data by using get dummies. Besides, there were some features that had values of N/A; we replaced them with the mean of their columns so that they don't influence the distribution.

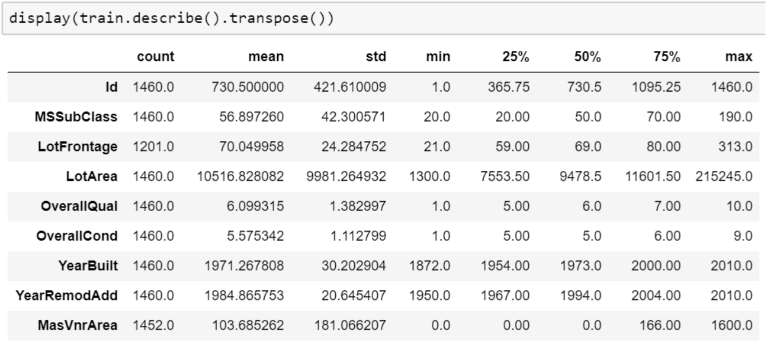
3. Models

We would perform two types of supervised learning algorithms: Extreme Gradient Boosting and Neural Network. XGBoost is used a algorithm to perform prediction because it is fast when compared to other implementations of gradient boosting. XGBoost dominates structured or tabular datasets on classification and regression predictive modeling problems. An extreme form of gradient boosting, XGBoost, is the preferred tool for Kaggle competitions due to its accuracy of predictions and speed. The quick speed in training a model is a result of allowing residuals to "roll-over" into the next tree. Careful selection of hyperparameters such as learning rate and max-depth, gave us better Kaggle prediction score. Fine tuning of the hyperparameters can be achieved through extensive cross-validation; however, caution is recommended when fitting too many parameters as it can be computationally expensive and time consuming.

4. Data Analysis and Data Processing

Data Analysis and visualization

Figure 1 shows the visualization of each feature of the dataset showing different type of classification of that feature.

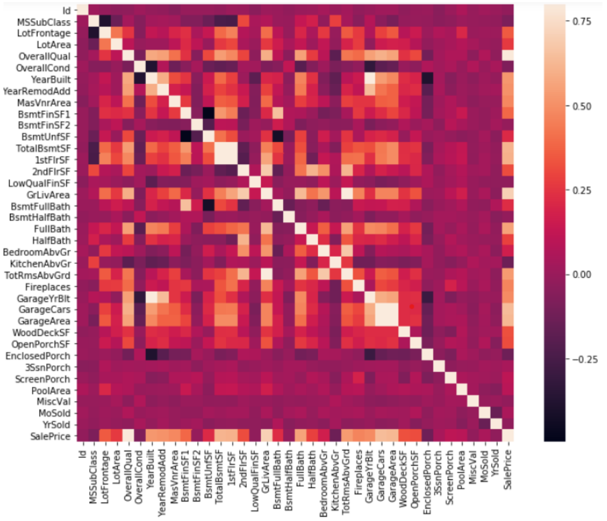


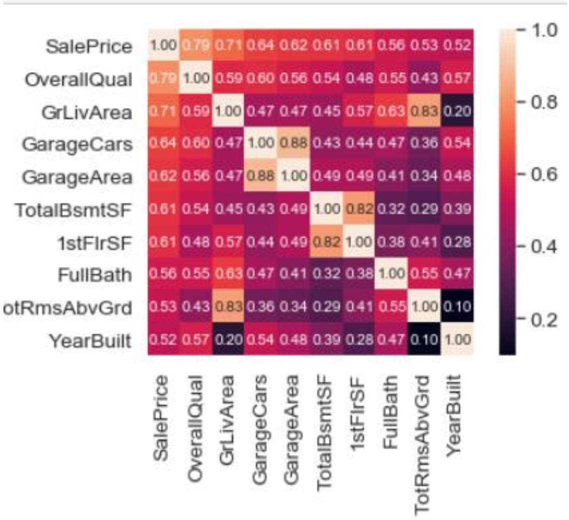
```
display(train.describe().transpose())
```

	count	mean	std	min	25%	50%	75%	max
Id	1460.0	730.500000	421.610009	1.0	365.75	730.5	1095.25	1460.0
MSSubClass	1460.0	56.897260	42.300571	20.0	20.00	50.0	70.00	190.0
LotFrontage	1201.0	70.049958	24.284752	21.0	59.00	69.0	80.00	313.0
LotArea	1460.0	10516.828082	9981.264932	1300.0	7553.50	9478.5	11601.50	215245.0
OverallQual	1460.0	6.099315	1.382997	1.0	5.00	6.0	7.00	10.0
OverallCond	1460.0	5.575342	1.112799	1.0	5.00	5.0	6.00	9.0
YearBuilt	1460.0	1971.267808	30.202904	1872.0	1954.00	1973.0	2000.00	2010.0
YearRemodAdd	1460.0	1984.865753	20.645407	1950.0	1967.00	1994.0	2004.00	2010.0
MasVnrArea	1452.0	103.685262	181.066207	0.0	0.00	0.0	166.00	1600.0

Fig 1.

Figure 2 Represents the co-relation between SalePrice and another feature of the dataset. GrLivArea is the lightest so it has highest co-relation with SalePrice.





	SalePrice	OverallQual	GrLivArea	GarageCars	GarageArea	TotalBsmtSF	1stFlrSF	FullBath	TotRmsAbvGrd	YearBuilt
SalePrice	1.00	0.79	0.71	0.64	0.62	0.61	0.61	0.56	0.53	0.52
OverallQual	0.79	1.00	0.59	0.60	0.56	0.54	0.48	0.55	0.43	0.57
GrLivArea	0.71	0.59	1.00	0.47	0.47	0.45	0.57	0.63	0.83	0.20
GarageCars	0.64	0.60	0.47	1.00	0.88	0.43	0.44	0.47	0.36	0.54
GarageArea	0.62	0.56	0.47	0.88	1.00	0.49	0.49	0.41	0.34	0.48
TotalBsmtSF	0.61	0.54	0.45	0.43	0.49	1.00	0.82	0.32	0.29	0.39
1stFlrSF	0.61	0.48	0.57	0.44	0.49	0.82	1.00	0.38	0.41	0.28
FullBath	0.56	0.55	0.63	0.47	0.41	0.32	0.38	1.00	0.55	0.47
otRmsAbvGrd	0.53	0.43	0.83	0.36	0.34	0.29	0.41	0.55	1.00	0.10
YearBuilt	0.52	0.57	0.20	0.54	0.48	0.39	0.28	0.47	0.10	1.00

Fig 2.1

Fig 2.2

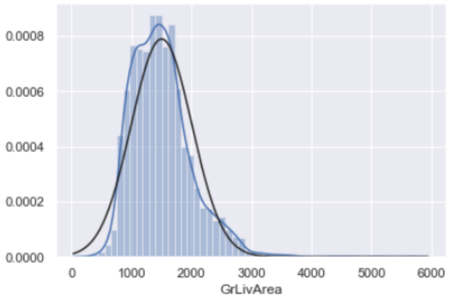


Fig 3.1

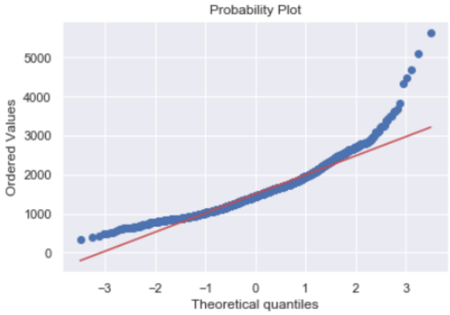


Fig 3.2

In figure 3 'SalePrice' is not normal. It shows 'peakedness', positive skewness and does not follow the diagonal line. But everything's not lost. A simple data transformation can solve the problem. In case of positive skewness, log transformations usually works well.

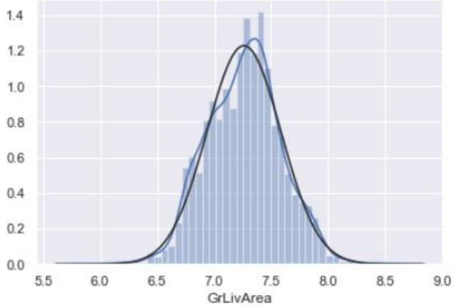


Fig 4.1

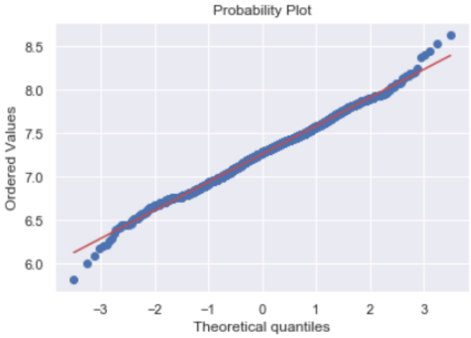


Fig 4.2

This how we have corrected the skewness of the feature “GrLivArea”

Data Processing for XGBoost

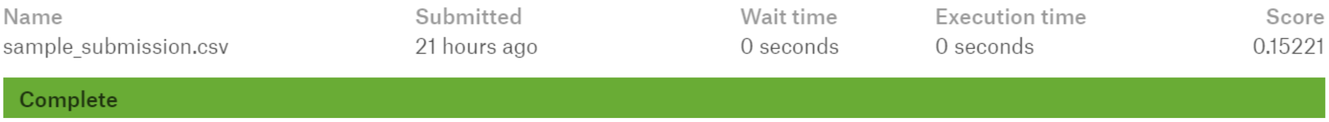
Handling missing value form the data to get better result from the model. Firstly we have tried to identify which column have large number of missing value which are almost more 60% we dropped those column from the dataset. Once we have done this part using heatmap we have identified the number of missing values left in the other columns of the dataset. As we see few of the columns still have some missing values left, we categories the numerical value feature and categorical value feature and perform mean and mode respectively depending on the type of the feature. We have identified the categorical columns from dataset and convert their categories into the numerical value using get_dummies. We also have seen number of categories in a same feature from train dataset and test dataset in different. So to handle this situation we concatenate the test data and train data and then do the get_dummies to get proper classification of categories to numeric conversion.

Data Processing for Artificial Neural Network

Similarly, we have handled the columns having data missing greater than 70%. Applied log transformation on a column “GrtLivArea” where we found the skewness in the probability plot. As the left missing value are handled in XGBoosst data processing same way we have handle the data in the Neural Net data processing. Encoded the categorical data using get_dummies function. Train and test dataset are scaled using StandardScalar from sklearn.

5. Model and Results

The two models used for this project are XGBoosting and Artificial Neural Network. Both the models work efficiently on regression data. XGBoosting provides good performance on sample size of up to 1000K whereas Neural Network can handle even bigger dataset size without getting saturated.



Name	Submitted	Wait time	Execution time	Score
sample_submission.csv	21 hours ago	0 seconds	0 seconds	0.15221

Complete

6. What we learned:

- Data Visualization technique helped us to understand the relation between all features.
- Data Processing is an important step which gives better performance.
- Understood how to determine outliers and process to remove them. How these outliers deviate from standard deviation.
- Heatmap is the best way to get a quick glance of the data and the relationship between different features.
- Missing data can imply a reduction of the sample size. This can prevent us from proceeding with the analysis.
- Outliers is also something that we should be aware of because outliers can markedly affect our models and can be a valuable source of information, providing us insights about specific behaviors.
- Normalization is a technique used to change the values of an array to a common scale, without distorting difference in the range of values. If we feed unnormalized data to neural network, the gradient will change differently for every column and thus the learning will oscillate.
- We have used XGBoost a regression model to predict the house pricing. In order to make the model perform better we can use hyper parameter optimization.