

|  |
| --- |
| AI Algorithms  F100 |
|  |
| **December 1,**2020  House Prices Prediction  Submitted by: Muhammad Imran (student # 100805394)  Fall 2020 |

Executive Summary

Artificial intelligence has become the breakout technology, utilizing huge amounts of computing power to learn and identify patterns in data.  Real estate industry has been driven by personal preferences and human interactions, primarily among buyers, sellers, and real estate agents. But now, changes to the buyer-agent-seller paradigm are happening. Real estate companies adopting Artificial Intelligence will be critical to sustaining and enhancing their competitive advantage to grow in this rapidly evolving industry.

Rationale Statement

Property price influences many different transactions such as sales and loan. Traditionally, property estimate is determined by professional. The risk is that human tend to be biased due to interest from lender, buyer, or seller. Therefore, a predicted system based on Artificial intelligence can serve as an independent and less biased system. For first time or less experienced buyers of real estate properties, an automated price prediction system can be useful to suggest underpriced or overpriced properties in the market.

## Data Acquisition

Data has been collected from Kaggle competition which is based on [Ames Housing dataset](http://www.amstat.org/publications/jse/v19n3/decock.pdf) , it was compiled by Dean De Cock for use in data science education.

## File descriptions

* **train.csv** - the training set
* **test.csv** - the test set

Description of the data in data file.

* **SalePrice** - the property's sale price in dollars. This is the target variable that you're trying to predict.
* **MSSubClass**: The building class
* **MSZoning**: The general zoning classification
* **LotFrontage**: Linear feet of street connected to property
* **LotArea**: Lot size in square feet
* **Street**: Type of road access
* **Alley**: Type of alley access
* **LotShape**: General shape of property
* **LandContour**: Flatness of the property
* **Utilities**: Type of utilities available
* **LotConfig**: Lot configuration
* **LandSlope**: Slope of property
* **Neighborhood**: Physical locations within Ames city limits
* **Condition1**: Proximity to main road or railroad
* **Condition2**: Proximity to main road or railroad (if a second is present)
* **BldgType**: Type of dwelling
* **HouseStyle**: Style of dwelling
* **OverallQual**: Overall material and finish quality
* **OverallCond**: Overall condition rating
* **YearBuilt**: Original construction date
* **YearRemodAdd**: Remodel date
* **RoofStyle**: Type of roof
* **RoofMatl**: Roof material
* **Exterior1st**: Exterior covering on house
* **Exterior2nd**: Exterior covering on house (if more than one material)
* **MasVnrType**: Masonry veneer type
* **MasVnrArea**: Masonry veneer area in square feet
* **ExterQual**: Exterior material quality
* **ExterCond**: Present condition of the material on the exterior
* **Foundation**: Type of foundation
* **BsmtQual**: Height of the basement
* **BsmtCond**: General condition of the basement
* **BsmtExposure**: Walkout or garden level basement walls
* **BsmtFinType1**: Quality of basement finished area
* **BsmtFinSF1**: Type 1 finished square feet
* **BsmtFinType2**: Quality of second finished area (if present)
* **BsmtFinSF2**: Type 2 finished square feet
* **BsmtUnfSF**: Unfinished square feet of basement area
* **TotalBsmtSF**: Total square feet of basement area
* **Heating**: Type of heating
* **HeatingQC**: Heating quality and condition
* **CentralAir**: Central air conditioning
* **Electrical**: Electrical system
* **1stFlrSF**: First Floor square feet
* **2ndFlrSF**: Second floor square feet
* **LowQualFinSF**: Low quality finished square feet (all floors)
* **GrLivArea**: Above grade (ground) living area square feet
* **BsmtFullBath**: Basement full bathrooms
* **BsmtHalfBath**: Basement half bathrooms
* **FullBath**: Full bathrooms above grade
* **HalfBath**: Half baths above grade
* **Bedroom**: Number of bedrooms above basement level
* **Kitchen**: Number of kitchens
* **KitchenQual**: Kitchen quality
* **TotRmsAbvGrd**: Total rooms above grade (does not include bathrooms)
* **Functional**: Home functionality rating
* **Fireplaces**: Number of fireplaces
* **FireplaceQu**: Fireplace quality
* **GarageType**: Garage location
* **GarageYrBlt**: Year garage was built
* **GarageFinish**: Interior finish of the garage
* **GarageCars**: Size of garage in car capacity
* **GarageArea**: Size of garage in square feet
* **GarageQual**: Garage quality
* **GarageCond**: Garage condition
* **PavedDrive**: Paved driveway
* **WoodDeckSF**: Wood deck area in square feet
* **OpenPorchSF**: Open porch area in square feet
* **EnclosedPorch**: Enclosed porch area in square feet
* **3SsnPorch**: Three season porch area in square feet
* **ScreenPorch**: Screen porch area in square feet
* **PoolArea**: Pool area in square feet
* **PoolQC**: Pool quality
* **Fence**: Fence quality
* **MiscFeature**: Miscellaneous feature not covered in other categories
* **MiscVal**: $Value of miscellaneous feature
* **MoSold**: Month Sold
* **YrSold**: Year Sold
* **SaleType**: Type of sale
* **SaleCondition**: Condition of sale

## 3. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an open-ended process where we calculate statistics and make figures to find trends, anomalies, patterns, or relationships within the data. The goal of EDA is to learn what our data can tell us. It generally starts out with a high-level overview, then narrows in to specific areas as we find intriguing areas of the data. The findings may be interesting in their own right, or they can be used to inform our modeling choices, such as by helping us decide which features to use.

## Approach

* Understand the problem. We'll look at each variable and do a philosophical analysis about their meaning and importance for this problem.
* Univariable study. We'll just focus on the dependent variable ('Sale Price') and try to know a little bit more about it.
* Multivariate study. We'll try to understand how the dependent variable and independent variables relate.
* Basic cleaning. We'll clean the dataset and handle the missing data, outliers and categorical variables.
* Test assumptions. We'll check if our data meets the assumptions required by most multivariate techniques.

#### 3.1 Data set

There are 1460 instances of training data and 1460 of test data. Total number of attributes equals 81, of which 36 is quantitative, 43 categorical + Id and SalePrice.

* Quantitative: 1stFlrSF, 2ndFlrSF, 3SsnPorch, BedroomAbvGr, BsmtFinSF1, BsmtFinSF2, BsmtFullBath, BsmtHalfBath, BsmtUnfSF, EnclosedPorch, Fireplaces, FullBath, GarageArea, GarageCars, GarageYrBlt, GrLivArea, HalfBath, KitchenAbvGr, LotArea, LotFrontage, LowQualFinSF, MSSubClass, MasVnrArea, MiscVal, MoSold, OpenPorchSF, OverallCond, OverallQual, PoolArea, ScreenPorch, TotRmsAbvGrd, TotalBsmtSF, WoodDeckSF, YearBuilt, YearRemodAdd, YrSold.
* Qualitative: Alley, BldgType, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, BsmtQual, CentralAir, Condition1, Condition2, Electrical, ExterCond, ExterQual, Exterior1st, Exterior2nd, Fence, FireplaceQu, Foundation, Functional, GarageCond, GarageFinish, GarageQual, GarageType, Heating, HeatingQC, HouseStyle, KitchenQual, LandContour, LandSlope, LotConfig, LotShape, MSZoning, MasVnrType, MiscFeature, Neighborhood, PavedDrive, PoolQC, RoofMatl, RoofStyle, SaleCondition, SaleType, Street, Utilities.
* **Train set has** 1460 rows & 81 Features-columns

Exploring Target Variable – Sales Price

descriptive statistics summary

count 1460.000000

mean 180921.195890

std 79442.502883

min 34900.000000

25% 129975.000000

50% 163000.000000

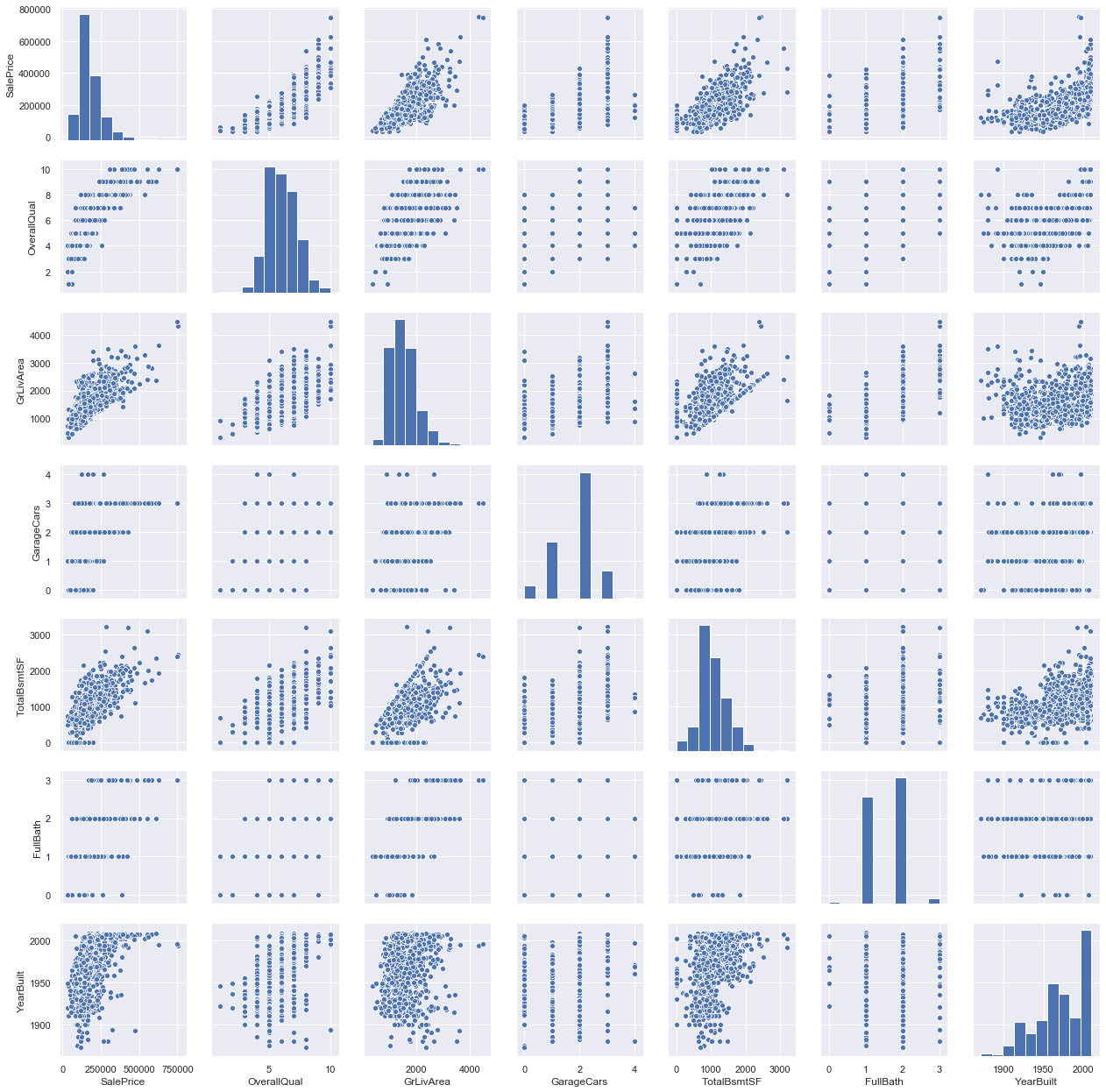
75% 214000.000000

max 755000.000000

## 

The target variable is right skewed. Linear models need normally distributed data , we need to transform this variable and make it more normally distributed.

3.2 Univariable study

****

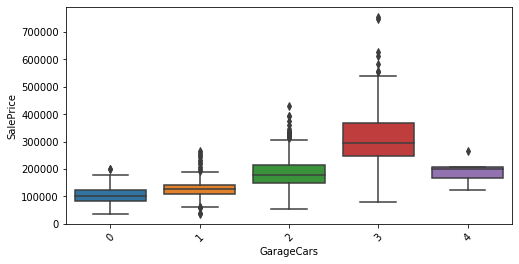
###### *TotalBsmtSF' and 'GrLiveArea' showing linear realation - drawing a linear line.*

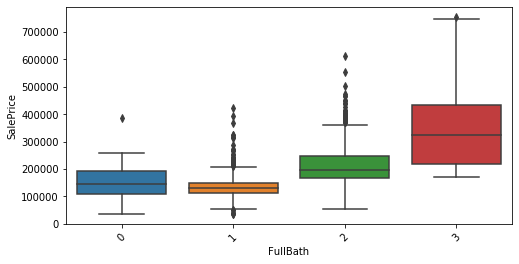
###### *SalePrice' and 'YearBuilt' showing booming real estate market.*

###### *In conclusion , these highly correlataed feature seems to be good contributor for predictions.*

Multivariate study

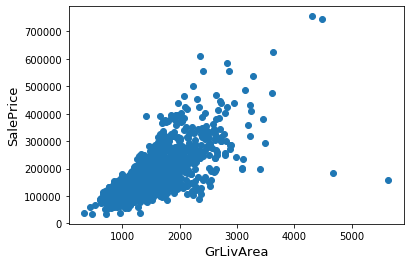
###### 

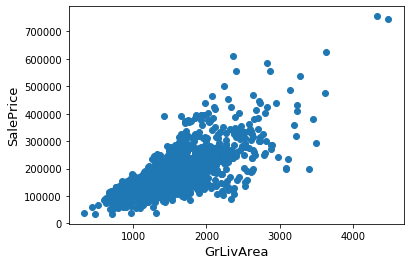
****

****

Based on expert recommendation, analysis of major features shows that features in above figures have positive relationship with target variable and hence would be a good contributor in predicting the sales price.

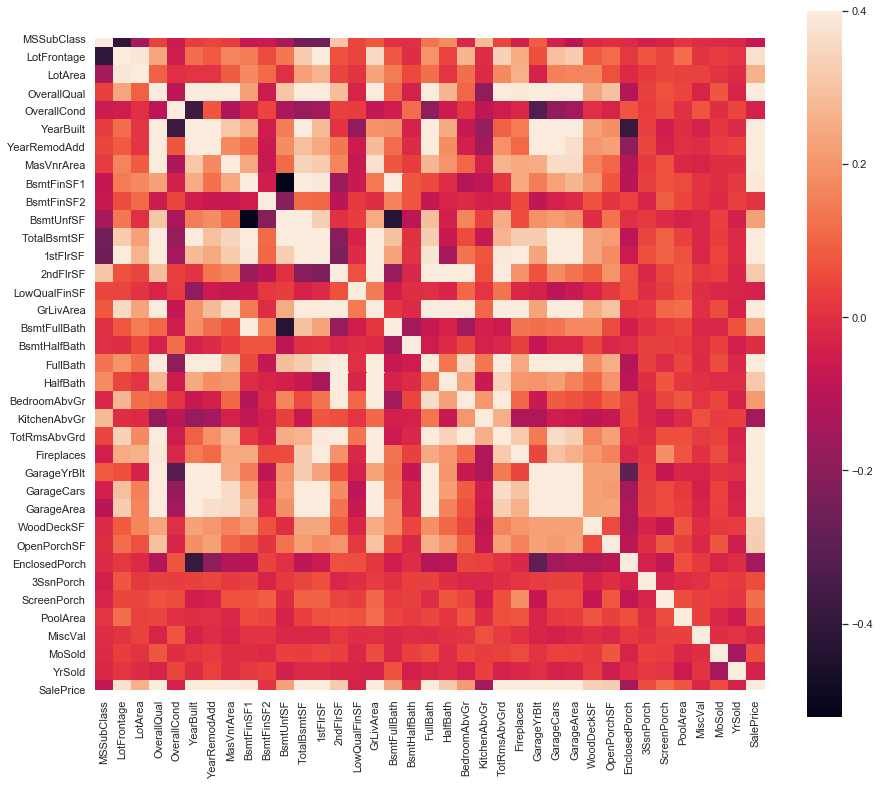
Outliers:

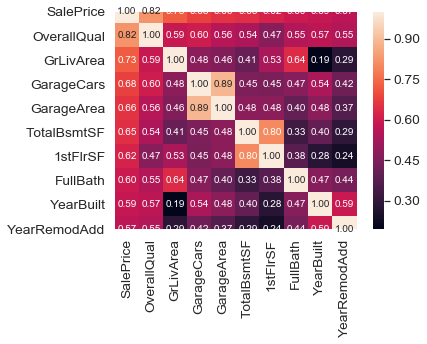
****

****

Outliers, GrLivArea > 4000 and sales price 300000 has been removed.

Correlation

****

****

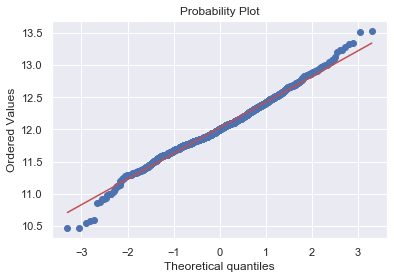
Heat Maps showing correlation between target variable and with other features. Though correlation is not necessarily tells the importance of independent features but it can provide a good idea for further analysis.

Top 10 Features showing highest correlation with sales are mentioned above figure.

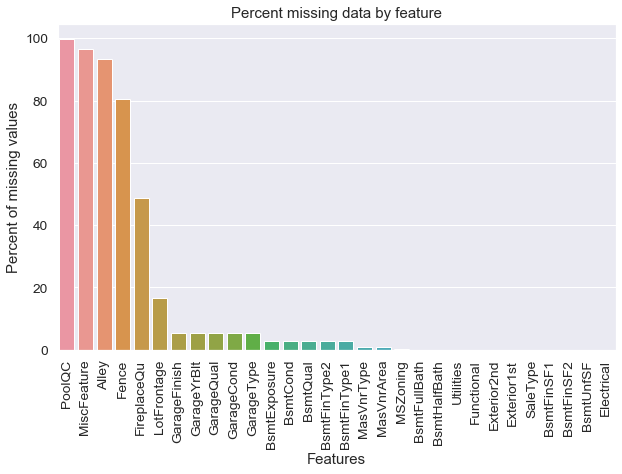
4. Data Processing

*SalesPrice distribution is not normal distribution,so before performing regression it has to be transformed.*

****

****

Missing Data



Handling Missing values

## Missing values replaced with none

all\_data[["Alley","Fence","FireplaceQu"]] = all\_data[["Alley","Fence","FireplaceQu"]].fillna("None")

## Replaced by median of Neighborhood

all\_data["LotFrontage"] = all\_data.groupby("Neighborhood")["LotFrontage"].transform(lambda x: x.fillna(x.median()))

## Replacing missing data with None

for col in ('GarageType', 'GarageFinish', 'GarageQual', 'GarageCond'):

all\_data[col] = all\_data[col].fillna('None')

## Replacing missing data with 0

for col in ('BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF','TotalBsmtSF', 'BsmtFullBath', 'BsmtHalfBath'):

all\_data[col] = all\_data[col].fillna(0)

for col in ('BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2'):

all\_data[col] = all\_data[col].fillna('None')

all\_data["MasVnrType"] = all\_data["MasVnrType"].fillna("None")

all\_data["MasVnrArea"] = all\_data["MasVnrArea"].fillna(0)

## 'RL' is by far the most common value. So we can fill in missing values with 'RL'

all\_data['MSZoning'] = all\_data['MSZoning'].fillna(all\_data['MSZoning'].mode()[0])

all\_data = all\_data.drop(['Utilities'], axis=1)

all\_data["Functional"] = all\_data["Functional"].fillna("Typ")

all\_data['Electrical'] = all\_data['Electrical'].fillna(all\_data['Electrical'].mode()[0])

all\_data['KitchenQual'] = all\_data['KitchenQual'].fillna(all\_data['KitchenQual'].mode()[0])

all\_data['Exterior1st'] = all\_data['Exterior1st'].fillna(all\_data['Exterior1st'].mode()[0])

all\_data['Exterior2nd'] = all\_data['Exterior2nd'].fillna(all\_data['Exterior2nd'].mode()[0])

all\_data['SaleType'] = all\_data['SaleType'].fillna(all\_data['SaleType'].mode()[0])

all\_data['MSSubClass'] = all\_data['MSSubClass'].fillna("None")

all\_data["MiscFeature"] = all\_data["MiscFeature"].fillna("None")

all\_data["PoolQC"] = all\_data["PoolQC"].fillna("None")

Transforming some numerical variables that are really categorical

all\_data['MSSubClass'] = all\_data['MSSubClass'].apply(str)

all\_data['OverallCond'] = all\_data['OverallCond'].astype(str)

all\_data['YrSold'] = all\_data['YrSold'].astype(str)

all\_data['MoSold'] = all\_data['MoSold'].astype(str)

5. Feature Engineering

##### Creation of new features from numerical features

**Combining some features based on the real life experience . This might help model to predict accurately. Also, there are other techniques like in some cases where linear relation is not able to separate different values, we use higher degree functions to find patterns**

all\_data['YearsSinceBuilt'] = all\_data['YrSold'].astype(int) - all\_data['YearBuilt']

all\_data['YearsSinceRemod'] = all\_data['YrSold'].astype(int) - all\_data['YearRemodAdd']

all\_data['OtherRooms'] = all\_data['TotRmsAbvGrd'] - all\_data['BedroomAbvGr'] - all\_data['KitchenAbvGr']

all\_data['TotalBathrooms'] = all\_data['FullBath'] + (0.5 \* all\_data['HalfBath']) + all\_data['BsmtFullBath'] + (0.5 \* all\_data['BsmtHalfBath'])

all\_data['LotDepth'] = all\_data['LotArea'] / all\_data['LotFrontage']

all\_data['TotalSF'] = all\_data['TotalBsmtSF'] + all\_data['1stFlrSF'] + all\_data['2ndFlrSF']

### Apply Box Cox Transformation of (highly) skewed features

* Apply encoding by creating dummy variables for categorical features.

Base Models:

* *LASSO Regression*

### Kernel Ridge Regression

### XGBoost

### LightGBM

Model Performance:

Evaluate how models perform on the data by evaluating the cross-validation rmsle error- root mean square log error.

Further Steps:

## Stacking models