Rahul Hunasehalli Rudranna Gowda, MSBA

# Machine Learning Final Project

# Team WSB

## Team members:

Rahul Hunasehalli Rudranna Gowda

Anvesh Puppala

Isaac Ratcliffe

## Business:

**EdLoan** is a financial services company in Ames, Iowa. They specialize in providing education loans to students in the community. The loans are usually collateralized by assets such as residentials properties. They have a data science team which is asked to come up with the prices of the houses in the city in order to validate the loan amounts requested. The qualitative and quantitative features of the houses are collated, analyzed and modelled in order to predict the house prices.

## Problem Statement:

The primary objective of the problem is to predict sale prices of individual residential properties in Ames, Iowa from 2006 to 2010. The training data set contains 1460 observations and many explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous; 80 in total) involved in assessing home values. The test dataset for which the predictions are made contains 1459 observations. The variables in the dataset focus on the quality and quantity of many physical attributes of the property. Most of the variables are exactly the type of information that a typical home buyer would want to know about a potential property (e.g., When was it built? How big is the lot? How many square feet of living space is in the dwelling? Is the basement finished? How many bathrooms are there?).

The Machine learning competition was based in **Kaggle.** The datasets were present in the webpage to download and work on. The code is written using Python 3.
A link to the competition is placed below:

https://www.kaggle.com/c/house-prices-advanced-regression-techniques

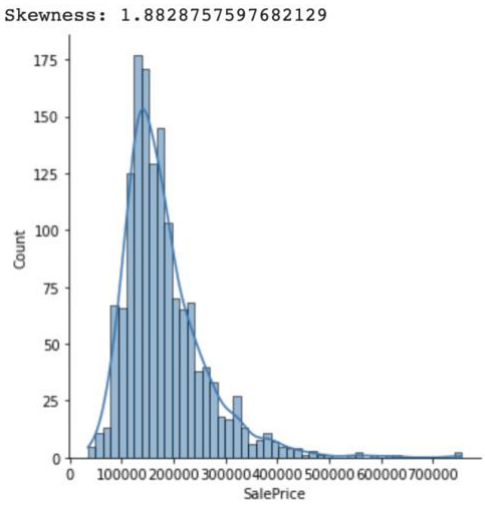
## Approach:

There were 2 datasets which were downloaded. One for the training data (train.csv) and another for the test data (test.csv). The various steps involved in building various regressors and coming up with predictions are mentioned below:

1. **Exploratory data Analytics (EDA):**

   A histogram of the dependent variable (*SalePrice*) is plotted in order to analyze the distribution of sale prices in the training dataset.

   The histogram is shown below in the following page.




We can see that; the histogram is right skewed. This means that the mean of the distribution is less than the median. Most of the properties are in the price range of 100,000 and 300,000. We can see a few properties which lie in the top end where prices go up to 700,000 and further. The skewness is equal to 1.88 which indicates high skewness.

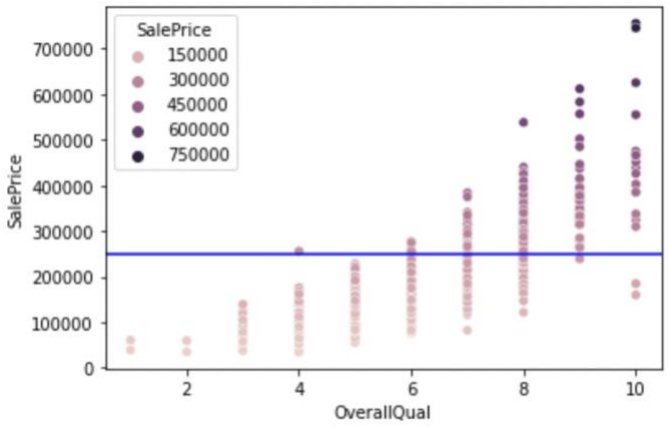
Next, the correlation between *SalePrice* and the other factors is calculated. The following screenshot shows the correlation numbers for the top 10 factors (descending order):

| | |
|---|---|
| SalePrice | 1.000000 |
| OverallQual | 0.790982 |
| GrLivArea | 0.708624 |
| GarageCars | 0.640409 |
| GarageArea | 0.623431 |
| TotalBsmtSF | 0.613581 |
| 1stFlrSF | 0.605852 |
| FullBath | 0.560664 |
| TotRmsAbvGrd | 0.533723 |
| YearBuilt | 0.522897 |

From the above correlation numbers, we can see that,

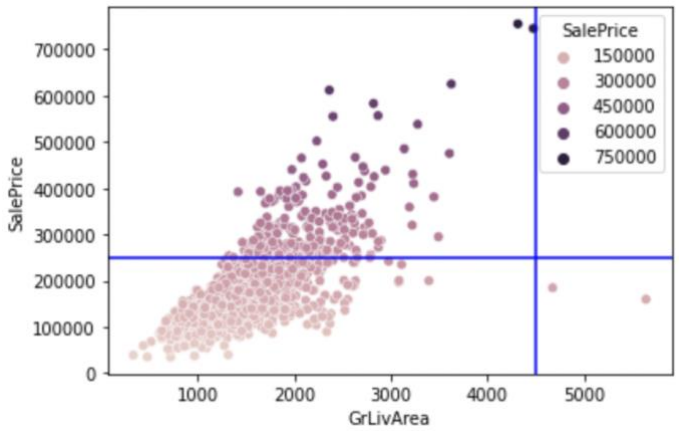
*OverallQual* (Rates the overall material and finish of the house) is the most correlated with *SalePrice* which is followed by *GrLivArea* (Above ground living area square feet).

In order to better understand the correlation, a scatterplot is plotted between *SalePrice* and *OverallQual*:



We can see that, there are 2 houses with *OverallQual* = 10 and which lie under the 250,000 *SalePrice* mark (blue line). These seem like outliers since, all the other houses with *OverallQual* = 10 start at 300,000.

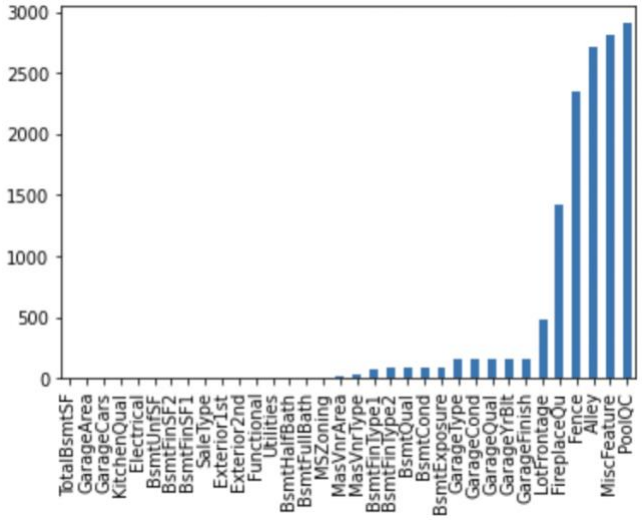
Further, another scatterplot between *SalePrice* and *OverallQual* is plotted:




We can see the same 2 outliers here as well, in the bottom right box. These outliers were removed from the dataset as they can influence our regression.

## 2. Missing data preparation:

A bar plot showing the missing data distribution among different columns is shown below:




We can see that there are a lot of missing values for the features *PoolQC, MiscFeature, Alley* and *Fence*. Hence, these features are dropped from the datasets.

Among the other features with missing values,

a) Numeric features: The missing values in numeric features are imputed with the median value present in the feature.
b) Non-numeric features: The missing values in non-numeric features are imputed with the mode of the feature.

## 3. Dummy Variables:

Dummy variables are created for the categorical features present in the datasets using the get_dummies() function within the pandas library, which uses one-hot encoding.

These dummy variables are appended to the final train dataset after removing the original categorical features. The same is done for the test dataset.

## 4. Modelling:

The training dataset is split into train and test subsets with 70% in the train subset.
Various regressors are used to fit the dataset.

1) Linear Regression
2) Random Forest Regressor
3) Lasso Regression
4) Decision Tree Regressor
5) XGBoost Regressor

The results of the above models are shown below:

The train_score (training R-Squared), test_score (testing R-Squared), train RMSE (training root-mean-squared-error) and test RMSE (testing root-mean-squared-error) values are tabulated and shown below:

| | model | train_score | test_score | train RMSE | test RMSE |
|---|---|---|---|---|---|
| **0** | XGBoost | 0.999126 | 0.932937 | 0.016716 | 0.121478 |
| **1** | Random Forest | 0.982677 | 0.889634 | 0.056405 | 0.146653 |
| **2** | Lasso | 0.941784 | 0.889072 | 0.105241 | 0.160656 |
| **3** | Linear Regression | 0.941784 | 0.888508 | 0.105249 | 0.160675 |
| **4** | Tree | 1.000000 | 0.794866 | 0.000000 | 0.204509 |

From the above results, we can see that **XGBoost** regressor performed the best among all the regressors with a test R-Squared of **0.933**. However, the interpretability of XGBoost regressor is low compared to LASSO and Linear regression. Another observation is that the train R-Squared of Decision Tree regressor is 1.0 (which means ideal or perfect model). However, the test R-Squared is the lowest among all the models (0.795), which indicates that the model **overfit** the dataset which is an expected characteristic of decision trees.

Finally, the test R-Squared values are plotted against each regressor and displayed as a bar chart: