CS 6375.501

MACHINE LEARNING FINAL PROJECT

FALL-2018

HOUSE PRICES: ADVANCED REGRESSION TECHNIQUES

TEAM MEMBERS:

KARAN KANANI (kyk170030)

TEJAS RAVI RAO (txr171830)

INSTRUCTOR

Prof. ANURAG NAGAR

TABLE OF CONTENTS

1. INTRODUCTION…….. 3
2. PROBLEM AND TECHNIQUE OVERVIEW……..... 3
3. RELATED BACKGROUND ……..... 4
4. DATASET DESCRIPTION……... 5
5. EXPERIMENTAL EVALUATION…….. 8
6. TECHNIQUES AND METHODS…….13
7. RESULTS AND ANALYSIS……15
8. CONCLUSION……. 16
9. REFERENCES…….. 16

1. INTRODUCTION

The prediction of Sale Price of a house usually depends on those factors that focus on the quality and quantity of many physical attributes of the property. These attributes or variables are exactly the type of information that a typical home buyer would want to know about a potential property. For example, the buyer would want to know the year in which the house was built? , or How many square feet of living is in the dwelling ?, or How many bathrooms are there? Etc.

This main purpose of this project is to apply various regression analysis techniques in predicting the Sale Price of the House. The project makes use of the Ames Housing Dataset complied by Dean De Cock. The dataset consists of 80 attributes that collectively decide the final Sale Price of the house.

The project is part of an ongoing Kaggle Competition where in participants submit their Final Sale Price Predictions. However, these submissions are evaluated on the Root-Mean-Squared-Error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price.

The main evaluation metric used in this project is based on Root-Mean-Squared-Error as mentioned earlier.

2. PROBLEM AND TECHNIQUE OVERVIEW

As mentioned earlier, the prediction of the Final Sale Price of the house is based on Advanced Regression Techniques. In this project, the techniques that will be applied to predict the target variable are namely Linear Regression, Bagging Regression, AdaBoost

Regression, Gradient Boosting Regression, Random Forest Regression and XGBoost Regression.

The 'SalePrice' attribute is transformed logarithmically in the dataset. Evaluation is done based on the Root-Mean-Squared-Error between the logarithm of predicted value and the logarithm of the sale price. The main reason being, applying a logarithmic transformation means that errors in predicting expensive houses and cheap houses will affect the result equally.

The project mainly aims at fitting each of the above-mentioned regression models to get required prediction values. The models will be trained on the training dataset with tuned parameters. The trained model is then run on a test dataset to predict the Sale Price (Logarithmic values). The predicted Sale price values are compared with the true Sale Price values based on RMSE values.

Additionally each model is also evaluated based on the Coefficient of determination denoted by R^2. The R^2 score determines the proportion of the variance in the dependent variable that is predictable from the independent variables. It provides a measure of how well observed outcomes are replicated and analyzed by the model, based on the proportion of total variation of outcomes explained by the model. The R^2 score has been determined for both the training dataset and test dataset for each of the 6 mentioned regression models.

3. RELATED BACKGROUND

3.1 Scikit – Learn

Scikit Learn is a free Machine Learning Library written in Python. It consists of various classification, regression and clustering algorithms and are designed to work in tandem with NumPy and SciPy. This library usually focusses on data modelling and not on the loading and manipulation of data.

3.2 Pandas

Pandas is a software library written in Python. Pandas is used for data manipulation and analysis. It mainly deals with multidimensional structural datasets and offers operations for manipulating numerical tables and time series.
The library provides DataFrame object for data manipulation with integrated indexing.

3.3 NumPy

NumPy is a free software library in Python. NumPy provides additional support for large multidimensional arrays and matrices. It also provides many high-level mathematical functions to operate on these structures.

3.4 Coefficient of Determination

The Coefficient of Determination, denoted by R^2 is the proportion of the variance in the dependent variable that is predictable from the independent variables. It provides a measure of how well observed outcomes are replicated by the model based on the proportion of total variation of outcomes explained by the model. The coefficient of determination ranges from 0 to 1.

3.5 Root-mean-square Error

RMSE is used to measure the difference between the values predicted by a model and the values observed. RMSE is a measure of accuracy, to compare errors of different models for a particular dataset. It is scale-dependent. RMSE is always non-negative and value of 0 would indicate perfect fit to the data. A lower RMSE value is always better compared to a higher one.

4. DATASET DESCRIPTION

The dataset consists of 79 explanatory variables describing almost every aspect of residential homes. The target variable is 'SalePrice'. The dataset used will be from 'train.csv' which contains 1460 instances of data.

Feature Description:

The following table describes the 79 features that influence the SalePrice

Feature Name	Feature Description
MSSubClass	Identifies the type of dwelling involved in the sale.
MSZoning	Identifies the general zoning classification of the sale.
LotFrontage	Linear feet of street connected to property
LotArea	Lot size in square feet
Street	Type of road access to property
Alley	Type of alley access to property
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 various conditions
Condition2	Proximity to various conditions (if more than one is present)
BldgType	Type of dwelling
HouseStyle	Style of dwelling
OverallQual	Rates the overall material and finish of the house
OverallCond	Rates the overall condition of the house
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	Evaluates the quality of the material on the exterior
ExterCond	Evaluates the present condition of the material on the exterior
Foundation	Type of foundation
BsmtQual	Evaluates the height of the basement
BsmtCond	Evaluates the general condition of the basement
BsmtExposure	Refers to walkout or garden level walls
BsmtFinType1	Rating of basement finished area
BsmtFinSF1	Type 1 finished square feet
BsmtFinType2	Rating of basement finished area (if multiple types)
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	Bedrooms above grade
Kitchen	Kitchens above grade
KitchenQual	Kitchen quality
TotRmsAbvGrd	Total rooms above grade
Functional	Home functionality
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 (MM)
YrSold	Year Sold (YYYY)
SaleType	Type of sale
SaleCondition	Condition of sale

5. EXPERIMENTAL EVALUATION

5.1 DATA PRE-PROCESSING

The dataset for the Houses Prices: Advanced Regression Techniques had 79 explanatory variables (almost) covering every aspect of residential homes in Ames, Iowa. Here is a step-by-step insight into the technique that we have used in data cleaning and pre-processing:

5.1.1 FEATURE ENGINEERING PROCESS

- **Handling Outliers**

The Documentation mentions about the training data outliers. We made an attempt to look at these outliers.

For example, For the feature '**GrLivArea'**, the scatter plot w.r.t. to '**SalePrice'** is:

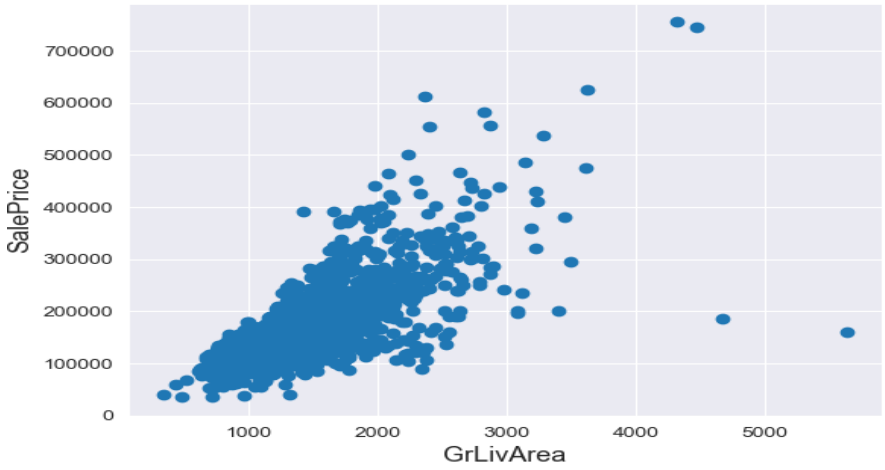

Figure 1: Two outliers spotted to the bottom right of the plot

The values with **GrLivArea** > 4000 with low **SalePrice** are outliers. Hence, we have removed them. Figure 2. Below show the two deleted outliers. Although there are other outliers in the training data, removing all of them may hinder the performance of the model.

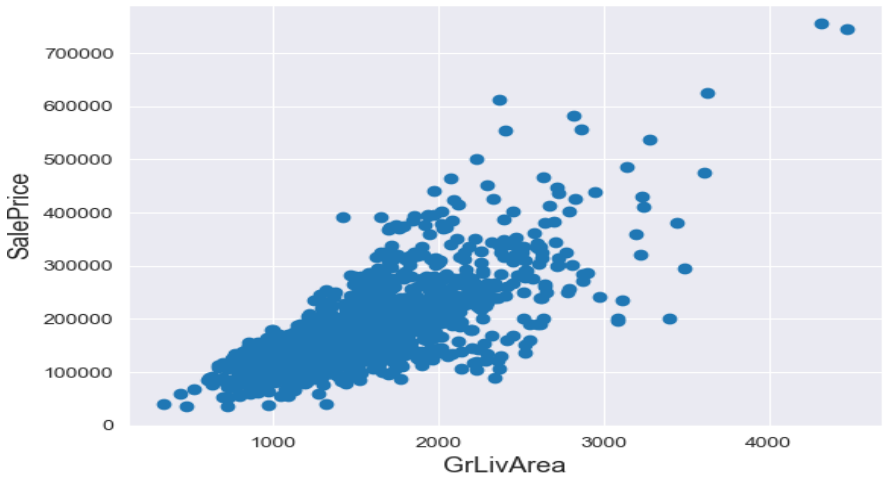

Figure 2: Two outliers present previously have been removed

- **Log-Transformation of the SalePrice (target variable)**

We observed that the **SalePrice** is skewed towards the right. It is prudent to convert this variable such that it is closer to normal distribution. Figure 3. Below shows a skewed distribution of the **SalePrice.**

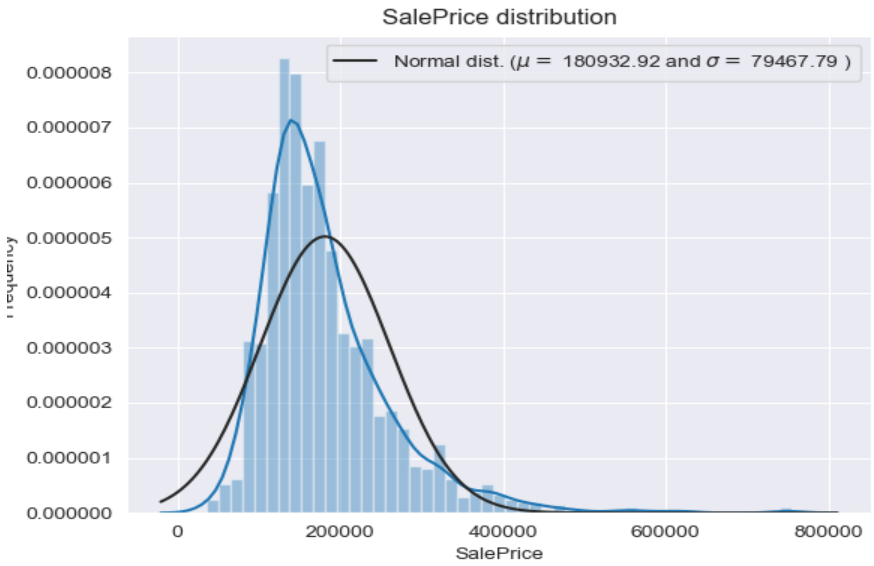

Figure 3: Right Skewed Distribution of SalePrice.

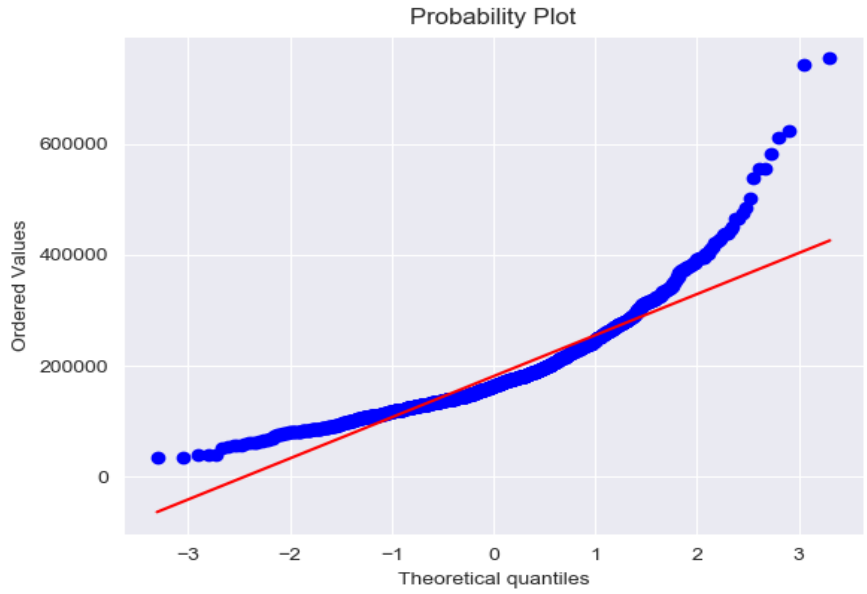

Figure 4: Probability Plot for Right Skewed SalePrice

Linear models function properly on normally distributed data. We need to transform this variable to make it normally distributed. We apply a Logarithmic Transformation on the SalePrice variable. Applying a logarithmic transformation means that errors in predicting expensive houses and cheap houses will affect the result equally. The figure below shows the distribution plot and probability plot after applying transformation.

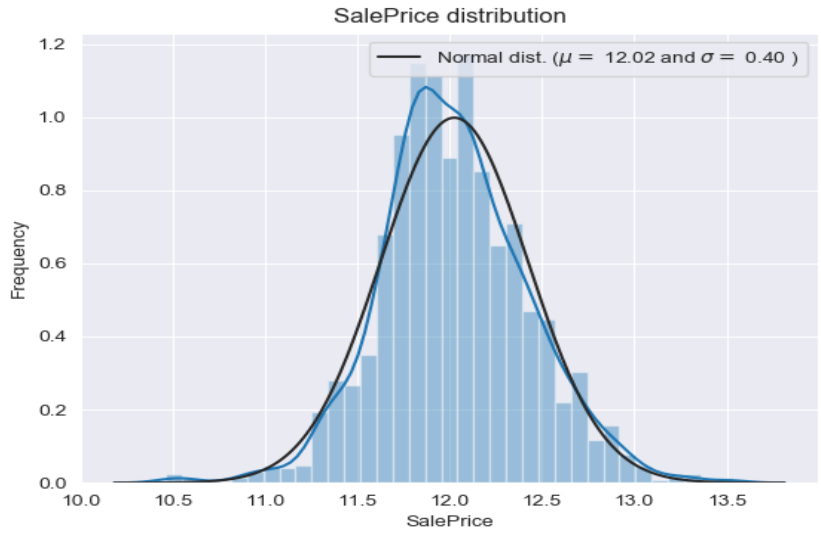

Figure 5: Normal Distribution of SalePrice.

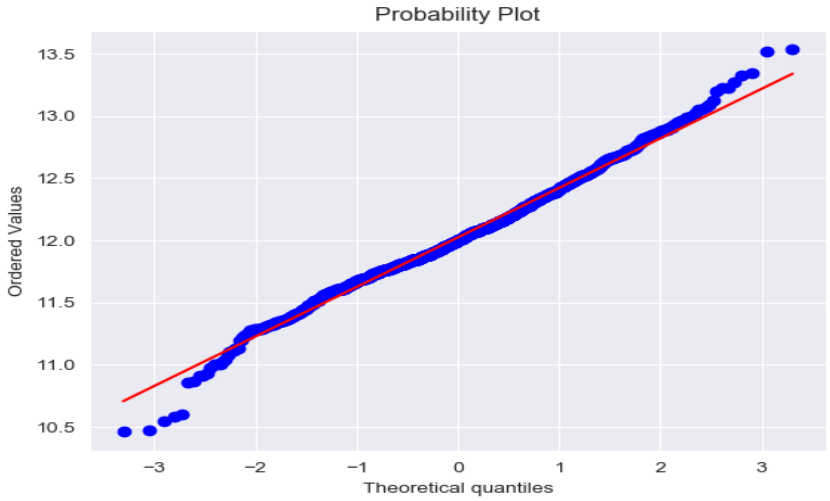

Figure 6: Probability plot shows a linear trend.

- **<u>Handling the missing data-values in a sequence</u>**

The following figure below shows the percentage of missing data by feature.

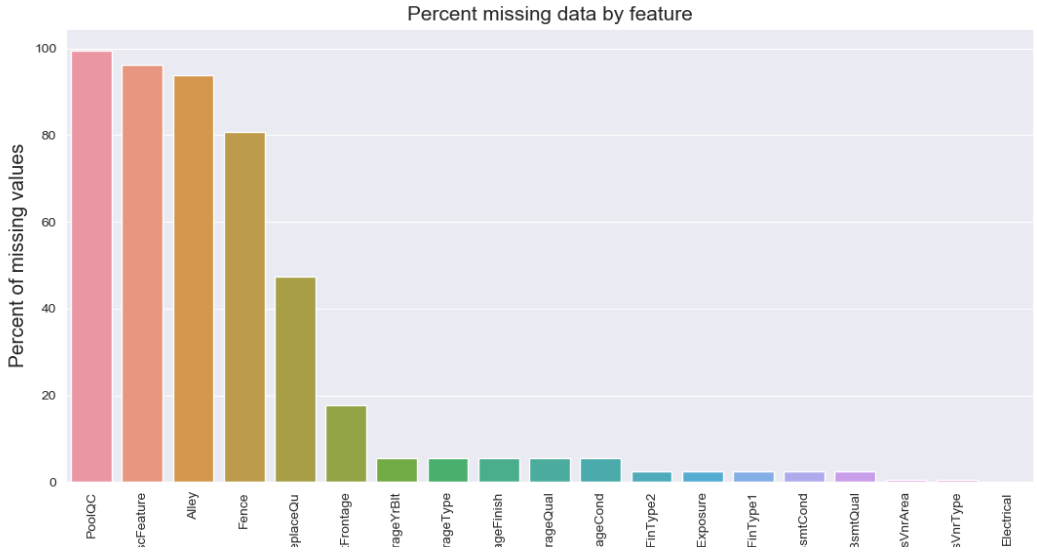

Figure 7: Percentage of missing data by feature

Before handling the missing values for the attributes, a heatmap correlation matrix is shown below. This helps us decide whether to keep the attribute with missing values or drop it entirely.

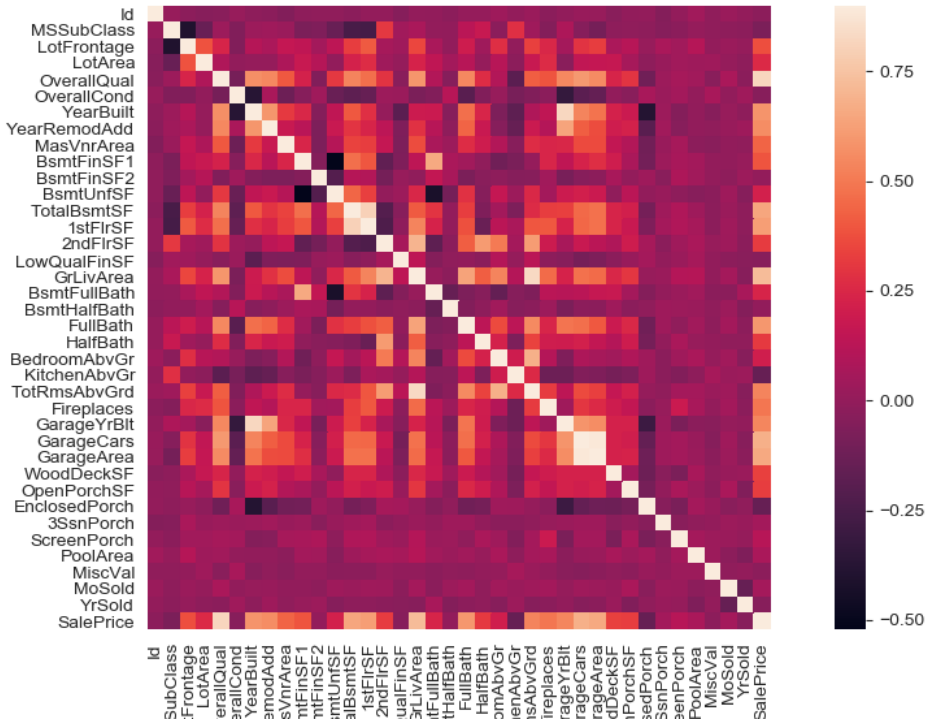

Figure 8: Correlation matrix of the attributes against SalePrice.

All the variables with missing values were handled such that some categorical variables with NA values were replaced with String 'None'. Others were replaced with the median. In the case of numerical features, some features were replaced with 0. In case of variables like **LotFrontage**, we applied group by on neighborhood and filled out the missing values by the median of that neighborhood. Any decision on how to replace the missing values for a column was only made after understanding the type of the feature and its values.

- **Changing the number features (which seem categorical) to String**

  There were some features like **MSSubClass, OverallCond, YrSold and MoSold** which have numbers as their values but are categorical in nature. Hence, we replaced them with the corresponding String for numbers only to convert them later on to numeric values by label encoder.

- **Encode labels for features with value between (0 and N classes -1)**

  Features like '**Fence', 'BsmtExposure', 'Street', 'Alley', 'CentralAir', 'OverallCond' etc.** were encoded using LabelEncoder.

6. PROPOSED TECHNIQUES

As the project expects the use of advanced regression techniques, the following Algorithms were implemented –

- Linear Regression
- Bagging Regression
- Random Forest Regression
- Gradient Boosting Regression
- AdaBoost Regression
- XGBoost Regression

Training and Validation:

For majority of the models mentioned above we have performed regression on two different split values

- With train-test split at 80%-20% for all models.
- With train-test split at 65%-35% for all models.

We made use of **GridSearch** to determine the best of the parameters to tune our models accordingly. The following table below shows the parameters used for each of the models.

Table:

Algorithm	Parameters
Linear Regressor	copy_X=True, fit_intercept=True, n_jobs=1, normalize=False
XGB Regressor	base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bytree=0.4603, gamma=0.0468, learning_rate=0.05, max_delta_step=0, max_depth=3, min_child_weight=1, missing=None, n_estimators=2500, n_jobs=1, nthread=None, objective='reg:linear', random_state=7, reg_alpha=0.464, reg_lambda=0.8571, scale_pos_weight=1, seed=None, silent=True, subsample=1
AdaBoost Regressor	base_estimator=None, learning_rate=0.05, loss='linear', n_estimators=500, random_state=None
Gradient Boosting Regressor	alpha=0.9, criterion='friedman_mse', init=None, learning_rate=0.05, loss='ls', max_depth=4, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=3000, presort='auto', random_state=5, subsample=1.0, verbose=0, warm_start=False
Random Forest Regressor	bootstrap=True, criterion='mse', max_depth=None, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, min_samples_leaf=1, min_samples_split=2, min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False
Bagging Regressor	base_estimator=None, bootstrap=True, bootstrap_features=False, max_features=1.0, max_samples=1.0, n_estimators=500, n_jobs=1, oob_score=False, random_state=None, verbose=0, warm_start=False

7. RESULTS AND ANALYSIS

As mentioned previously, regression was performed on two different values of train-test split.

- With train-test split at 80%-20% for all models the following results were obtained

Model	R^2 Score for train	R^2 Score for test	Root Mean Squared Error
Linear Regressor	0.919	0.922	0.115
XGB Regressor	0.949	0.926	0.112
AdaBoost Regressor	0.855	0.850	0.159
Gradient Boosting Regressor	1.000	0.936	0.105
Random Forest Regressor	0.983	0.915	0.120
Bagging Regressor	0.982	0.915	0.120

- With train-test split at 65%-35% for all models the following results were obtained.

Model	R^2 Score for train	R^2 Score for test	Root Mean Squared Error
Linear Regressor	0.925	0.902	0.126
XGB Regressor	0.953	0.902	0.126
AdaBoost Regressor	0.870	0.823	0.169
Gradient Boosting Regressor	1.000	0.914	0.118
Random Forest Regressor	0.983	0.888	0.135
Bagging Regressor	0.983	0.887	0.135

The main evaluation criteria used here are the Coefficient of Determination (R^2) and Root-Mean-Square Error. R^2 Score gives a measure of how well observed outputs are replicated by the tuned model, based on the proportion of total variation of outputs determined by the model. Root-Mean-Squared Error (RMSE) is a measure of the difference between the predicted SalePrice and the observed SalePrice. A lower RMSE is better as compared to a higher one.

The following models performed well compared to other models for both split values.

- Gradient Boosting Regression model
- Linear Regression model

It is evident from the above two observation tables that the R^2 values are higher compared to those of other models. Similarly, we also find the RMSE values of these 2 models to be lower than the other 4 models. The Prediction of **SalePrice** (Although in Logarithmic Value) is better in Gradient Boosting Regression Model and Linear Regression Model.

8. CONCLUSION

This project mainly helped us in identifying the importance of feature engineering and pre-processing. The above-mentioned pre-processing methods helped us in fitting a tuned regression model to the data to make a prediction with lesser errors.

The project involved making use of 6 regression techniques namely Linear Regression, XGBoost Regression, Gradient Boosting Regression, AdaBoost Regression, Bagging Regression and Random Forest Regression. We find that **Gradient Boosting Regression** model performed better compared to other models in predicting the SalePrice. This was evaluated based on R^2 score and RMSE value.

9. REFERENCES

- https://www.kaggle.com/pmarcelino/comprehensive-data-exploration-with-python
- https://www.kaggle.com/serigne/stacked-regressions-top-4-on-leaderboard
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html
- https://xgboost.readthedocs.io/en/latest/python/python_api.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html