Capstone Final Presentation (Team 1)

IOWA – House Prediction

Team: Johnson Henry Ravikumar

Mentor: Dr. Narayana , Anwesh Reddy

Problem Statement

One might wonder what drives the price of a house? Is it the neighbourhood? The size of the house? The amenities? Or something else?

The study was to find the best algorithm to predict the house prices in Iowa by focusing on reducing RMSE and increased R^2 score

More on the dataset:

The Ames Housing dataset was compiled by Dean De Cock and is commonly used in data science education, it has 1460 observations with 79 explanatory variables in train dataset describing (almost) every aspect of residential homes in Ames, Iowa. The test data comprises of 1459 observations with 79 explanatory variables.

Approach:

Below are the steps I followed to get the ideal RMSE

1. Exploratory Data Analysis (EDA)
2. Data cleaning
    1. Outlier removal
    2. Missingness imputation
    3. Dummification
3. Pre Model
4. Cross-Validation (Hyperparameter tuning)
5. Final Model

Exploratory Data Analysis (EDA)

Categorization:

I started by exploring and understanding the dataset. I divided our variables into categories: Nominal Categorical, Ordinal Categorical and Target variable.

	A	B
1	**Nomial**	**Ordinal**
2	MSZoning	Street
3	LandContour	Alley
4	Utilities	LotShape
5	LotConfig	LandSlope
6	Neighborhood	ExterQual
7	Condition1	ExterCond
8	Condition2	BsmtQual
9	BldgType	BsmtCond
10	HouseStyle	BsmtExposure
11	RoofStyle	BsmtFinType1
12	RoofMatl	BsmtFinType2
13	Exterior1st	HeatingQC
14	Exterior2nd	CentralAir
15	MasVnrType	KitchenQual
16	Foundation	Functional
17	Heating	FireplaceQu
18	Electrical	GarageFinish
19	GarageType	GarageCond
20	GarageQual	PavedDrive
21	MiscFeature	Fence
22	SaleType	PoolQC
23	SaleCondition	

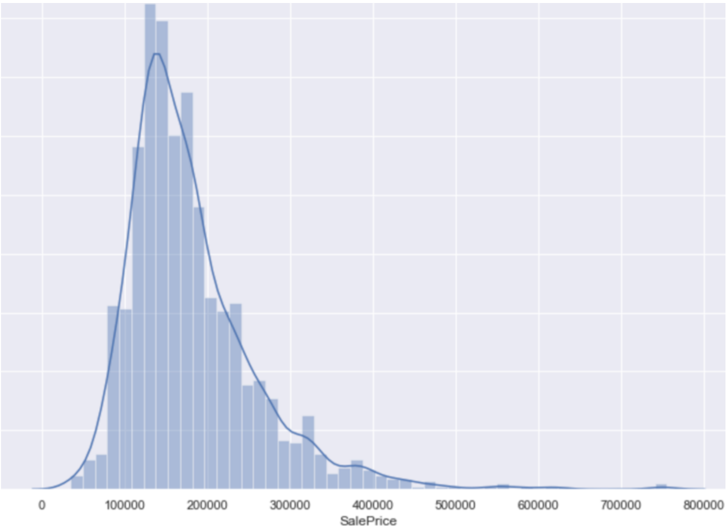
The above figure represents the analysis of segregating the features. I have done this by understanding the features by looking at the description given and the values.

Target Variable:

Sale price is the value we are looking to predict in our project, so it made sense to examine this variable first. The sale price exhibited a right-skewed distribution that was corrected by taking the log. Once the log was taken, we were no longer violating the normality assumption for regressions.

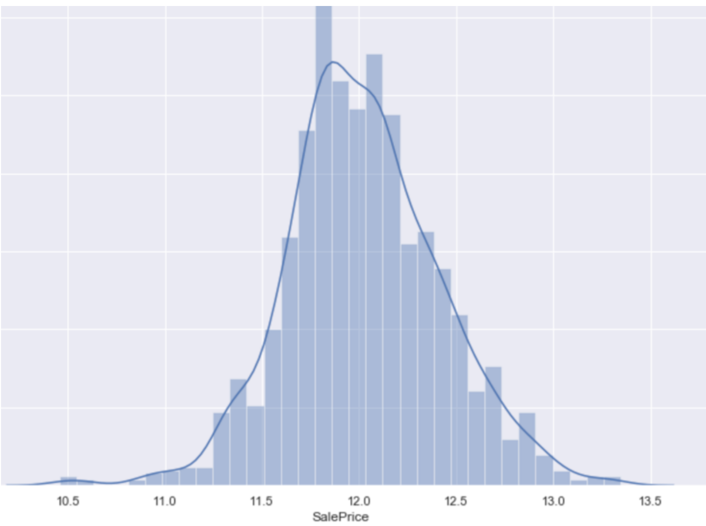
Sale Price before log transformation:

The distribution of price is right-skewed. We will use Log1p transformation technique to make the distribution more normal.



Sale Price after log transformation:

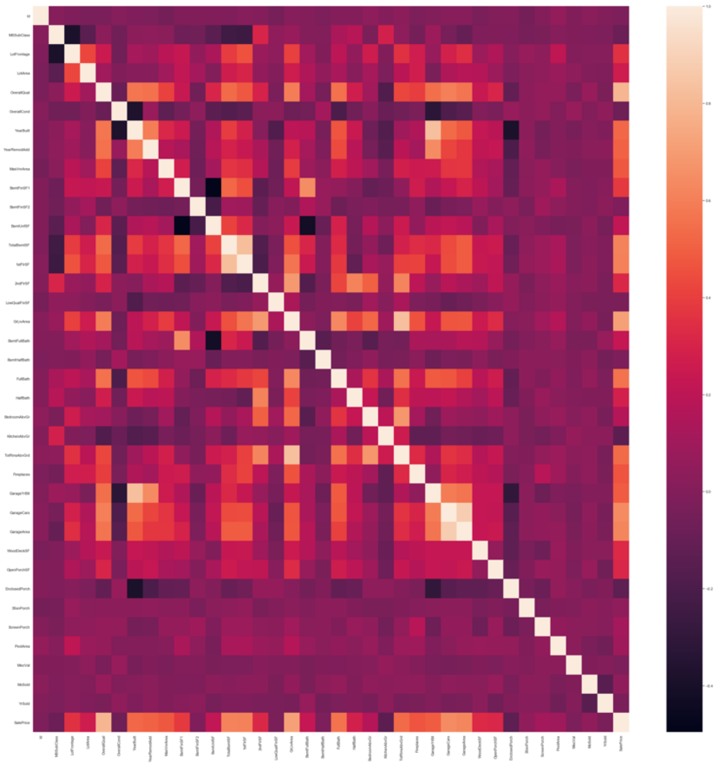
After the log1p transformation, the price distribution looks much more linearly distributed.



I have verified that this transformation can help improve the linear model's performance.

Correlation Levels:

Another graphical view of our analysis is the correlation plot that indicates levels of correlation amongst continuous variables, and between continuous features and the response variable (SalePrice).



In the above figure the last column on the x axis is the sale price . You can see the features with lighter shade that is highly correlated with sale price.

It definitely aided in the exploration of the data. I found that the Sale Price is strongly correlated with these continuous variables, so focused on finding outliers from these predictors.

Predictor: Correlation with Price

OverallQual : 0.790982

GrLivArea : 0.708624

GarageCars : 0.640409

GarageArea: 0.623431

TotalBsmtSF : 0.613581

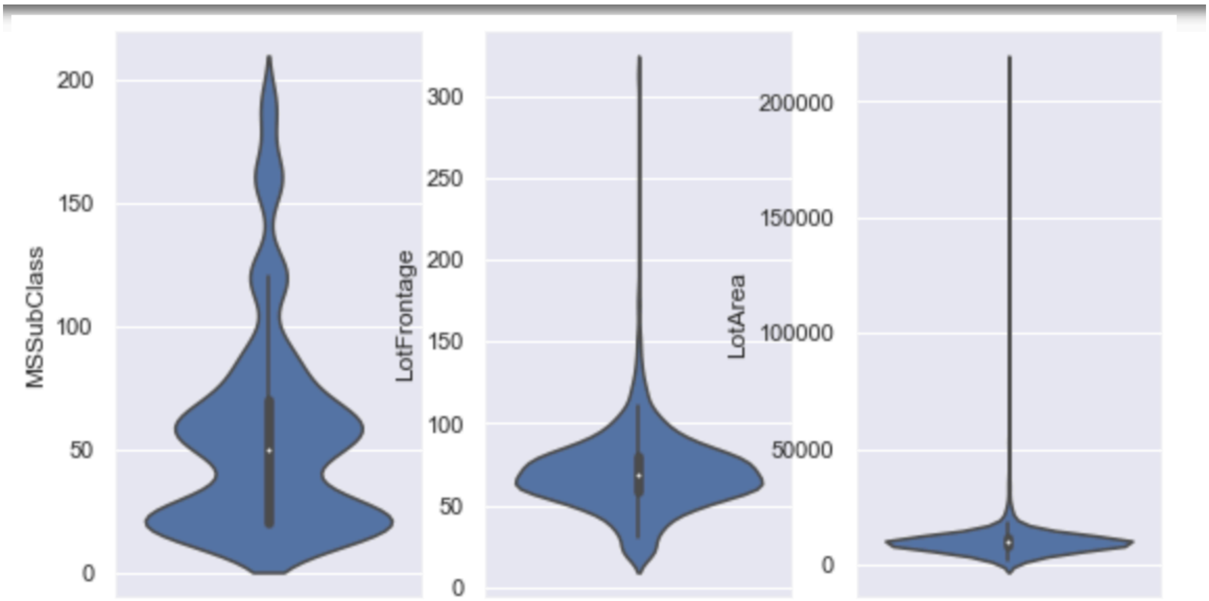
1stFlrSF : 0.605852

TotRmsAbvGrd : 0.533723

FullBath : 0.560664

Data Distribution visualisation

I wanted to see the distribution of data to check the skewness and this time I used Violin plot instead of box plots as its gives the density along with the distribution which helps in better visualisation.



Above fig is a sample violin plot across 2 distributions.. You can see the density and distribution.

Missing data

At One point while doing EDA I was checking on missing data and I could find 3 feature missing more than 93% of the data. I tried dropping the ones with more than 99% data missing but it negatively impacted the RMSE. Here is a split of which features missed more data

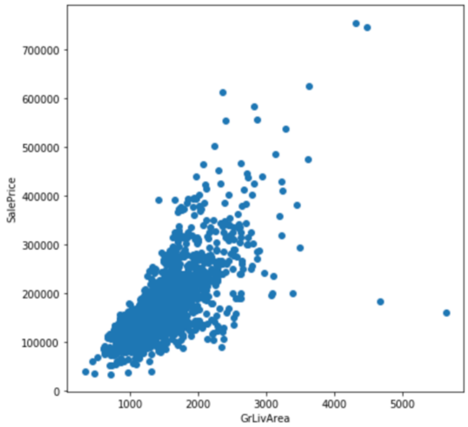
	total	% data missing
PoolQC	1453	99.520548
MiscFeature	1406	96.301370
Alley	1369	93.767123
Fence	1179	80.753425
FireplaceQu	690	47.260274
LotFrontage	259	17.739726
GarageType	81	5.547945
GarageCond	81	5.547945
GarageFinish	81	5.547945
GarageQual	81	5.547945
GarageYrBlt	81	5.547945
BsmtFinType2	38	2.602740
BsmtExposure	38	2.602740
BsmtQual	37	2.534247
BsmtCond	37	2.534247
BsmtFinType1	37	2.534247
MasVnrArea	8	0.547945
MasVnrType	8	0.547945
Electrical	1	0.068493

Data cleaning

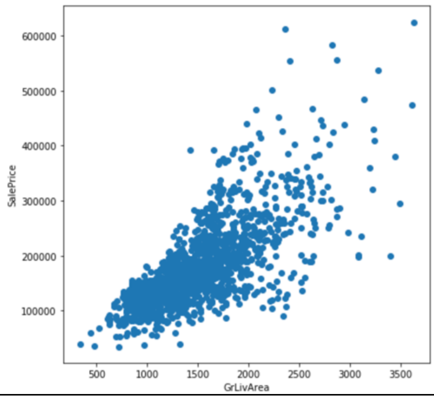
Outlier removal

In the next step I focused on Outlier removal. I took the predictors we identified in the correlation step which had very strong correlation with target variable and used the scatter plot to check on any outliers. I could find ‘GrLivArea’ had few outliers and I removed them as it might impact the model performance

Before removal



After removal



Missingness and Imputation:

Next, I decided to look at missing values by feature in the train and test dataset. As you can see below, there was significant missingness by feature across both these datasets. Most of the missing data corresponded to the absence of a feature. For example, the GarageType and MiscFeature showed up as "NA" if the house did not have a Garage or (Elevator or Tennis court) . These were imputed as “No Garage” or “No Feature” depending on the feature type before I did dummies for the remaining categorical values.

Logarithm Tranformation

After checking for skewness, I identified features having skewness greater than 0.75 and applied Log1p transformation. Following plot helped us in visualizing and identifying them

Before Log1p transformation

```
Id                 0.001342
MSSubClass         1.406366
LotFrontage        1.536435
LotArea           12.587561
OverallQual        0.183871
OverallCond        0.690631
YearBuilt         -0.610087
YearRemodAdd      -0.499831
MasVnrArea         2.648987
BsmtFinSF1         0.744855
BsmtFinSF2         4.248587
BsmtUnfSF          0.921759
TotalBsmtSF        0.486395
1stFlrSF           0.867081
2ndFlrSF           0.777866
LowQualFinSF       8.998564
GrLivArea          0.835192
BsmtFullBath       0.591152
BsmtHalfBath       4.128967
FullBath           0.017694
HalfBath           0.684223
BedroomAbvGr       0.215067
KitchenAbvGr       4.481366
TotRmsAbvGrd       0.661416
Fireplaces         0.632678
GarageYrBlt       -0.645821
GarageCars        -0.343475
GarageArea         0.132991
WoodDeckSF         1.551271
OpenPorchSF        2.339846
EnclosedPorch      3.084454
3SsnPorch         10.289866
ScreenPorch        4.115641
PoolArea          17.522613
MiscVal           24.443364
MoSold             0.217883
YrSold             0.093214
SalePrice          1.565959
```

After Log1p transformation

```
]:  Id                 0.001342
    MSSubClass         0.252130
    LotFrontage       -0.987033
    LotArea           -0.180382
    OverallQual        0.183871
    OverallCond        0.690631
    YearBuilt         -0.610087
    YearRemodAdd      -0.499831
    MasVnrArea         0.506128
    BsmtFinSF1         0.744855
    BsmtFinSF2         2.518748
    BsmtUnfSF         -2.182650
    TotalBsmtSF        0.486395
    1stFlrSF          -0.004520
    2ndFlrSF           0.295677
    LowQualFinSF       7.449565
    GrLivArea         -0.113799
    BsmtFullBath       0.591152
    BsmtHalfBath       3.956185
    FullBath           0.017694
    HalfBath           0.684223
    BedroomAbvGr       0.215067
    KitchenAbvGr       3.863121
    TotRmsAbvGrd       0.661416
    Fireplaces         0.632678
    GarageYrBlt       -0.675158
    GarageCars        -0.343475
    GarageArea         0.132991
    WoodDeckSF         0.159343
    OpenPorchSF       -0.020274
    EnclosedPorch      2.107656
    3SsnPorch          7.723866
    ScreenPorch        3.144867
    PoolArea          17.006205
    MiscVal            5.162852
    MoSold             0.217883
    YrSold             0.093214
    SalePrice          0.065460
```

Model:

We applied 6 different models on our data sets as follows:

- Linear model
- Ridge model
- Lasso model
- Elastic model
- Decision Tree model

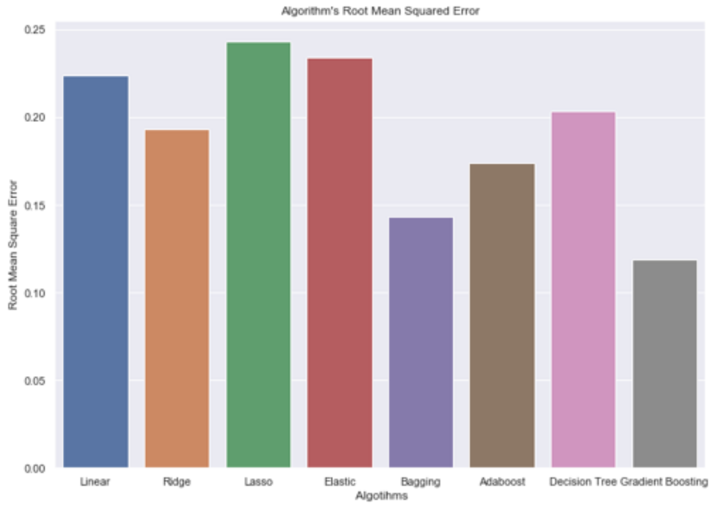
And following ensemble techniques

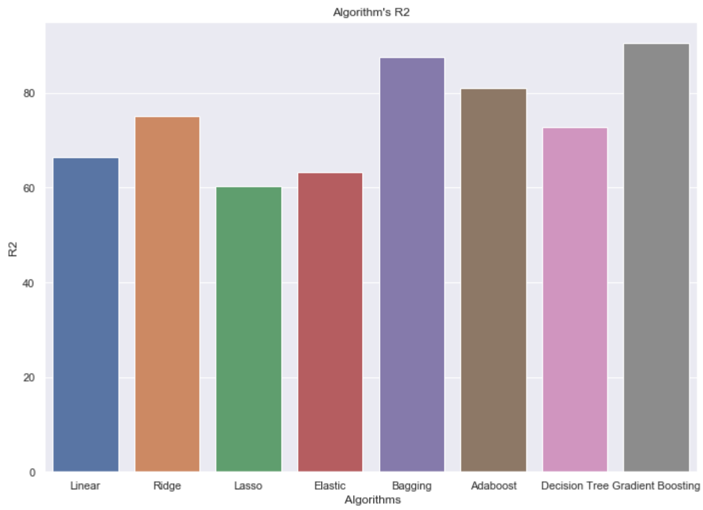
- Bagging
- Ada Boosting
- Gradient Boosting

All models used grid search cross-validation function to find the ideal parameters

In the pre-modeling phase, the train data set have been further divided into training and testing data sets, therefore, I was able to calculate RMSE and R^2 without any EDA or Feature Engineer and below are the results.

Model Performance before EDA		
Model	**RMSE**	**R^2**
Linear	0.2236	66.37
Lasso	0.1929	74.97
Ridge	0.243	60.28
ElasticNet	0.2338	63.21
Bagging	0.1429	87.43
AdaBoost	0.1739	80.91
Decision Tree	0.2031	72.71
Gradient Boosting	0.119	90.47

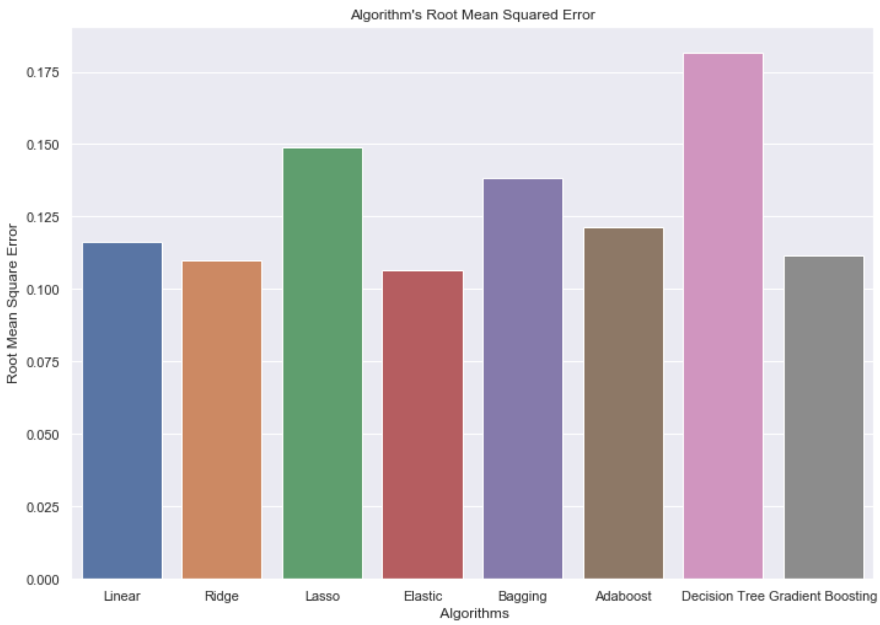


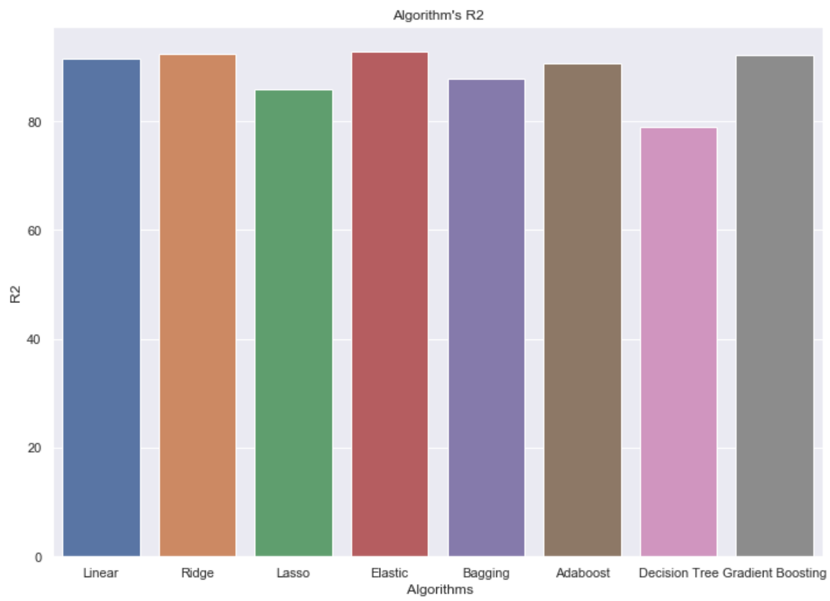


The premodelling result shows that Gradient Boosting performed well with a RMSE of **0.1190** and R^2 of **90.47**

After this I tried applying the features one by one and checking the RMSE and ran grid search for hyper parameter tuning and finally found an improved RMSE of **0. 1097** and R^2 of **92.77**

Model Performance after EDA		
Model	**RMSE**	**R^2**
Linear	0.1162	91.39
Lasso	0.1097	92.33
Ridge	0.1498	85.87
ElasticNet	0.1065	92.77
Bagging	0.1383	87.78
AdaBoost	0.1241	90.64
Decision Tree	0.1814	79.01
Gradient Boosting	0.1117	92.04





If you compare our final results with our pre modeling result, you can see RMSEs have decreased for all the models which is a clear indication the model is not overfitting

The **Elastic Net** model has the lowest RMSE as **0.1103** and highest R^2 at **92.77**. In contrast, the **Decision Tree model** has the worst RMSE as **0.1814** and R^2 score of **79**

Conclusion

It was a great learning overall. For data cleaning and imputation, the most important thing was to identify the categorical variables and numeric variables. The variable like MS SubClass is a numerical data type, but it actually is a categorical variable.

From what I have observed I could say, Linear models tend to outperform tree-based in terms of speed and score but it also depends on what EDA you incorporate and it drastically changes each model

It would be interesting to have tried out feature engineering by adding new featuring using a combination of exiting feature which positively influence the target but in the limited time couldn't explore it, but would definitely try it out later..