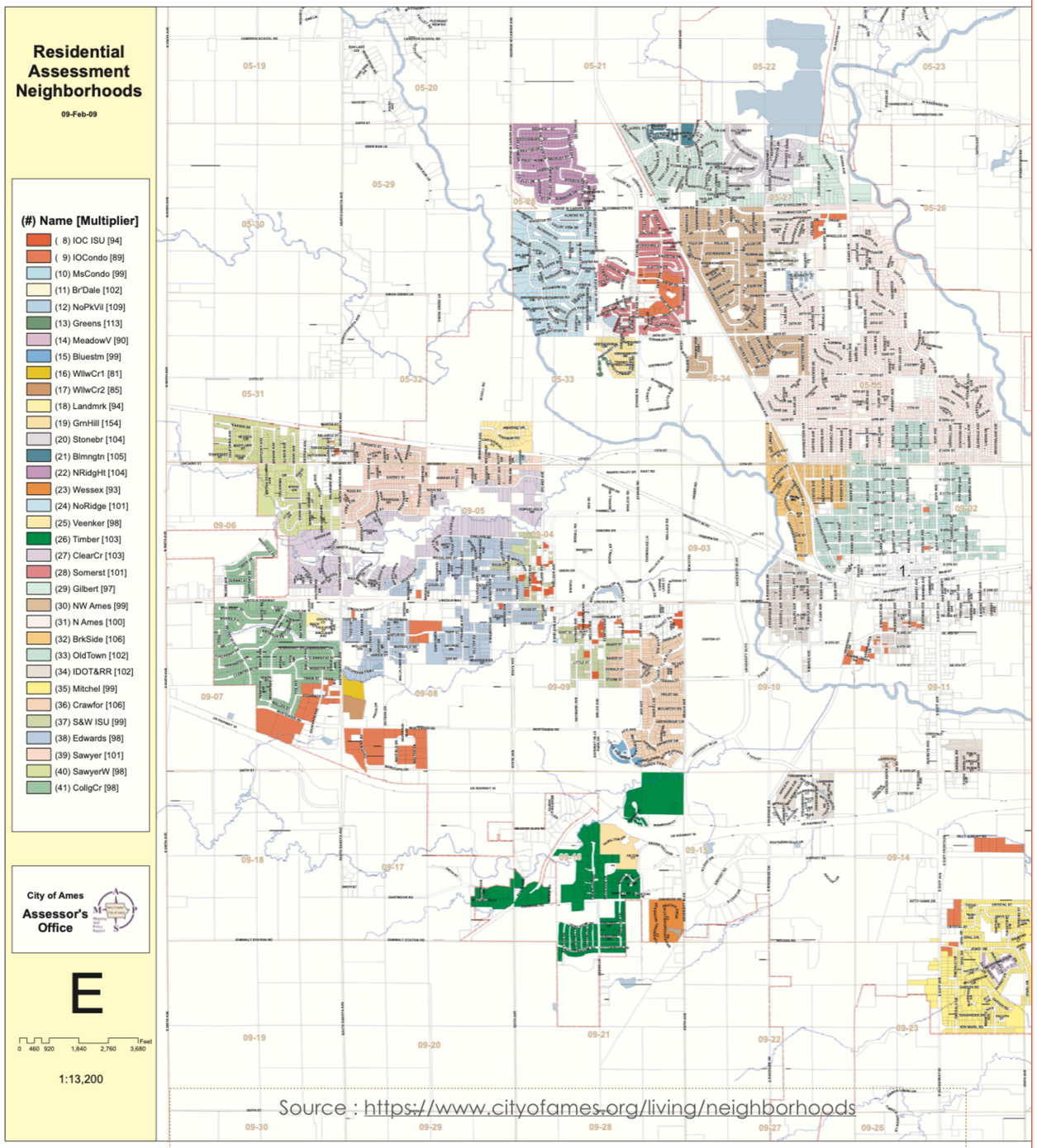
AMES IOWA HOUSE PRICE PREDICTION

RTA - Teik Ning Yang



Source : https://www.cityofames.org/living/neighborhoods

Kaggle
Predicting House Prices with Advanced Regression Techniques

GOAL

1) Create an effective price prediction model
2) Validate the model's prediction accuracy
3) Identify the important home price attributes which feed the model's predictive power.

METRIC

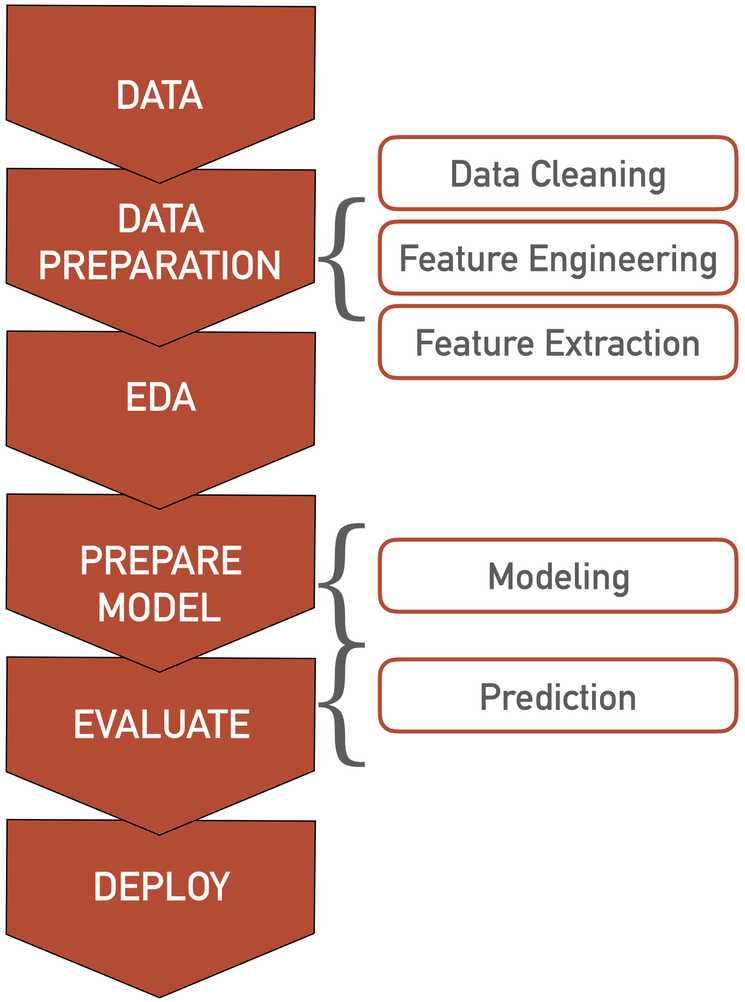
Minimised the Root-Mean-Squared-Error (RMSE) between the logarithm of the predicted value and the logarithm of the observed sales price.

ACKNOWLEDGEMENT

The Ames Housing dataset was compiled by Dean De Cock for use in data science education.

DATASET DETAILS

These data sets has 79 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, it requires a representative model to predict the final price of each home.



Kaggle
Predicting House Prices with Advanced Regression Techniques

DataSet

These data sets compresses of 2 sets of datas:
1.Train data (1460 instances X 79 explanatory features, 1 ID & 1 target)
2,Test data (1459 instances X 79 explanatory features, 1 ID)

```
Train dataset has 1460 rows and 81 columns
Test dataset has 1459 rows and 80 columns
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

Combine both Train and Test Dataset for Data Processing

Total number of features in total dataset which has Nan = 34

	NAN	Type	%
PoolQC	2909	object	99.657417
MiscFeature	2814	object	96.402878
Alley	2721	object	93.216855
Fence	2348	object	80.438506
FireplaceQu	1420	object	48.646797
LotFrontage	486	float64	16.649538
GarageFinish	159	object	5.447071
GarageQual	159	object	5.447071
GarageCond	159	object	5.447071
GarageYrBlt	159	float64	5.447071
GarageType	157	object	5.378554
BsmtExposure	82	object	2.809181
BsmtCond	82	object	2.809181
BsmtQual	81	object	2.774923
BsmtFinType2	80	object	2.740665
BsmtFinType1	79	object	2.706406
MasVnrType	24	object	0.822199
MasVnrArea	23	float64	0.787941
MSZoning	4	object	0.137033
BsmtFullBath	2	float64	0.068517
BsmtHalfBath	2	float64	0.068517
Functional	2	object	0.068517
Utilities	2	object	0.068517
GarageArea	1	float64	0.034258
GarageCars	1	float64	0.034258
Electrical	1	object	0.034258
KitchenQual	1	object	0.034258
TotalBsmtSF	1	float64	0.034258
BsmtUnfSF	1	float64	0.034258
BsmtFinSF2	1	float64	0.034258
BsmtFinSF1	1	float64	0.034258
Exterior2nd	1	object	0.034258
Exterior1st	1	object	0.034258
SaleType	1	object	0.034258

Kaggle
redicting House Prices with Advanced Regression Techniques

Data Preparation

Data Cleaning

For Train Data, there are 19 columns of features with either "Nan" values or missing data
For Test Data, there are 33 columns of features with either "Nan" values or missing data

Numeric Variables
There are 36 relevant numerical features.
Mainly stated the Sizes (sqf), Quantities of , Condition Rating and Time
Note : MSSubClass - is categorical coded as numeric

Categorical Variables
There are 43 relevant features.
Mainly stated the Zoning, Features, Materials and Condition Rating

Data pre-processing summary
Sale Price (Target) : Log Transform the target variable
Features : All features went thru Max-Min Scalar

Activity	No of Features
Data clean and coding	79 —> 211
Feature Engineering : YrSold, MoSold -> Sold Fr2016	211—> 210
Feature Selection : multicollinearity >80%	210—> 190
Feature Selection : Variance Threshold 100%	190
Feature Selection : Lasso / RFECV	190 —> 101

Treatment for Both Numeric and Categorical Data

Feature(Numeric)	Observation	Action
LotFrontage	Nan should be missing data.	'Nan' replace with mean
GarageYrBlt	Compare to GarageArea, it is clear than Nan is No Garage	Replace 'Nan' with mean
MasVnrArea	Nan should be missing data	'Nan' replace by mean
BsmtFullBath	Nan should be missing data	'Nan' replace by mean
BsmtHalfBath	Nan should be missing data	'Nan' replace by mean
GarageArea	Nan should be missing data	'Nan' replace by mean
TotalBsmtSF	Nan should be missing data	'Nan' replace by mean
BsmtUnfSF	Nan should be missing data	'Nan' replace by mean
BsmtFinSF2	Nan should be missing data	'Nan' replace by mean
BsmtFinSF1	Nan should be missing data	'Nan' replace by mean

Feature(Categorical)	Observation	Action
PoolQC	Compare to PoolArea, it is clear than Nan is representing No Pool	Change 'Nan' to 'Ne'
MiscFeature	From File Description, it is stated that NA is None (No Feature)	Change 'Nan' to 'Ne'
Alley	From File Description, it is stated that NA is No Alley	Change 'Nan' to 'Ne'
Fence	From File Description, it is stated that NA is No Fence	Change 'Nan' to 'Ne'
FireplaceQu	Compare to Fireplaces, it is clear than Nan is representing No Fireplace	Change 'Nan' to 'Ne'
GarageType	Compare to GarageArea, it is clear than Nan is No Garage	Change 'Nan' to 'Ne'
GarageFinish	Compare to GarageArea, it is clear than Nan is No Garage	Change 'Nan' to 'Ne'
GarageQual	Compare to GarageArea, it is clear than Nan is No Garage	Change 'Nan' to 'Ne'
GarageCond	Compare to GarageArea, it is clear than Nan is No Garage	Change 'Nan' to 'Ne'
BsmtExposure	Compare to TotalBsmtSF, it is clear than Nan is No Basement	Change 'Nan' to 'Ne'
BsmtFinType2	Compare to TotalBsmtSF, it is clear than Nan is No Basement	Change 'Nan' to 'Ne'
BsmtFinType1	Compare to TotalBsmtSF, it is clear than Nan is No Basement	Change 'Nan' to 'Ne'
BsmtCond	Compare to TotalBsmtSF, it is clear than Nan is No Basement	Change 'Nan' to 'Ne'
BsmtQual	FCompare to TotalBsmtSF, it is clear than Nan is No Basement	Change 'Nan' to 'Ne'
MasVnrType	Obsersation have Nan and None, which both could be same.	Change 'Nan','None' to 'Ne'
Electrical	There is one Nan, likely Missing Value	replace 'Nan' with mode
MSZoning	There is 4 Nan in test data, likely Missing Value	replace 'Nan' with mode
Functional	Nan , likely Missing Value	replace 'Nan' with mode
Utilities	Nan , likely Missing Value	replace 'Nan' with mode
GarageCars	Nan , likely Missing Value (numeric data but categorical bynature)	replace 'Nan' with mode
KitchenQual	Nan , likely Missing Value	replace 'Nan' with mode
Exterior2nd	Nan , likely Missing Value	replace 'Nan' with mode
Exterior1st	Nan , likely Missing Value	replace 'Nan' with mode
SaleType	Nan , likely Missing Value	replace 'Nan' with mode

Kaggle
redicting House Prices with Advanced Regression Techniques

Data Preparation

Data Cleaning

For Train Data, there are 19 columns of features with either "Nan" values or missing data
For Test Data, there are 33 columns of features with either "Nan" values or missing data

Numeric Variables
There are 36 relevant numerical features.
Mainly stated the Sizes (sqf), Quantities of , Condition Rating and Time
Note : MSSubClass - is categorical coded as numeric

Categorical Variables
There are 43 relevant features.
Mainly stated the Zoning, Features, Materials and Condition Rating

Data pre-processing summary
Sale Price (Target) : Log Transform the target variable
Features : All features went thru Max-Min Scalar

Activity	No of Features
Data clean and coding	79 —> 211
Feature Engineering : YrSold, MoSold -> Sold Fr2016	211—> 210
Feature Selection : multicollinearity >80%	210—> 190
Feature Selection : Variance Threshold 100%	190
Feature Selection : Lasso / RFECV	190 —> 101

Kaggle - Predicting House Prices with Advanced Regression Techniques

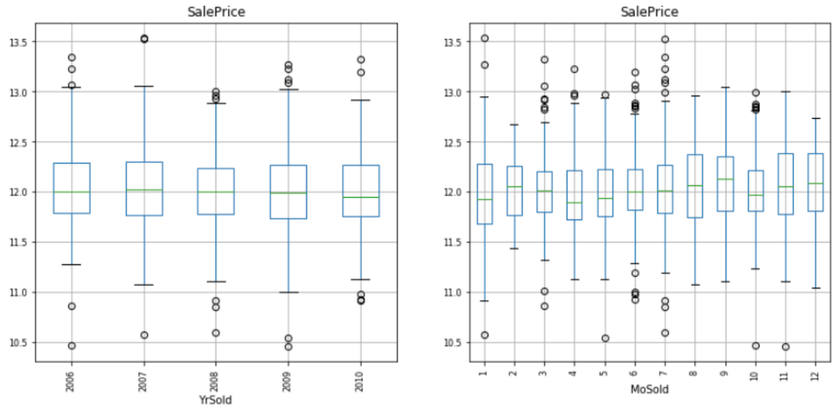
SalePrice

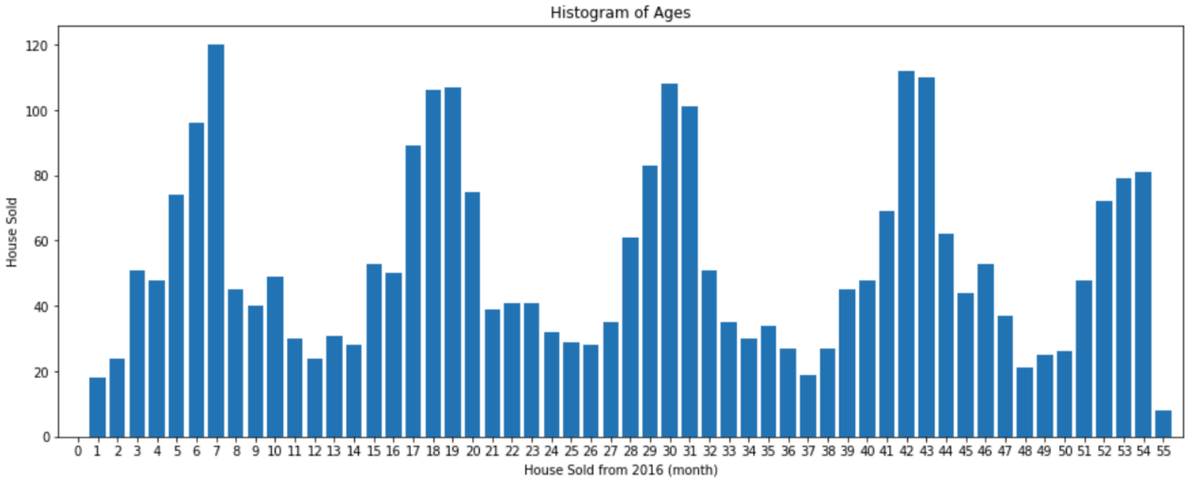
Taking logs means that errors in predicting expensive houses and cheap houses will affect the result equally.

EDA

Many numeric type features seems not well correlated to Sale Price, such as ‘Year Sold’, ‘Mouth Sold’.
However, there are several variables demonstrates some extend of correlation with sale price.
These variables generally related to size of the house (area), quantity of house’s feature (no. of bedroom, garage etc) and year built or renovate of the house. Most obvious correlation was observed for ‘Overall Qual’ and ‘ GrlivAreas’.

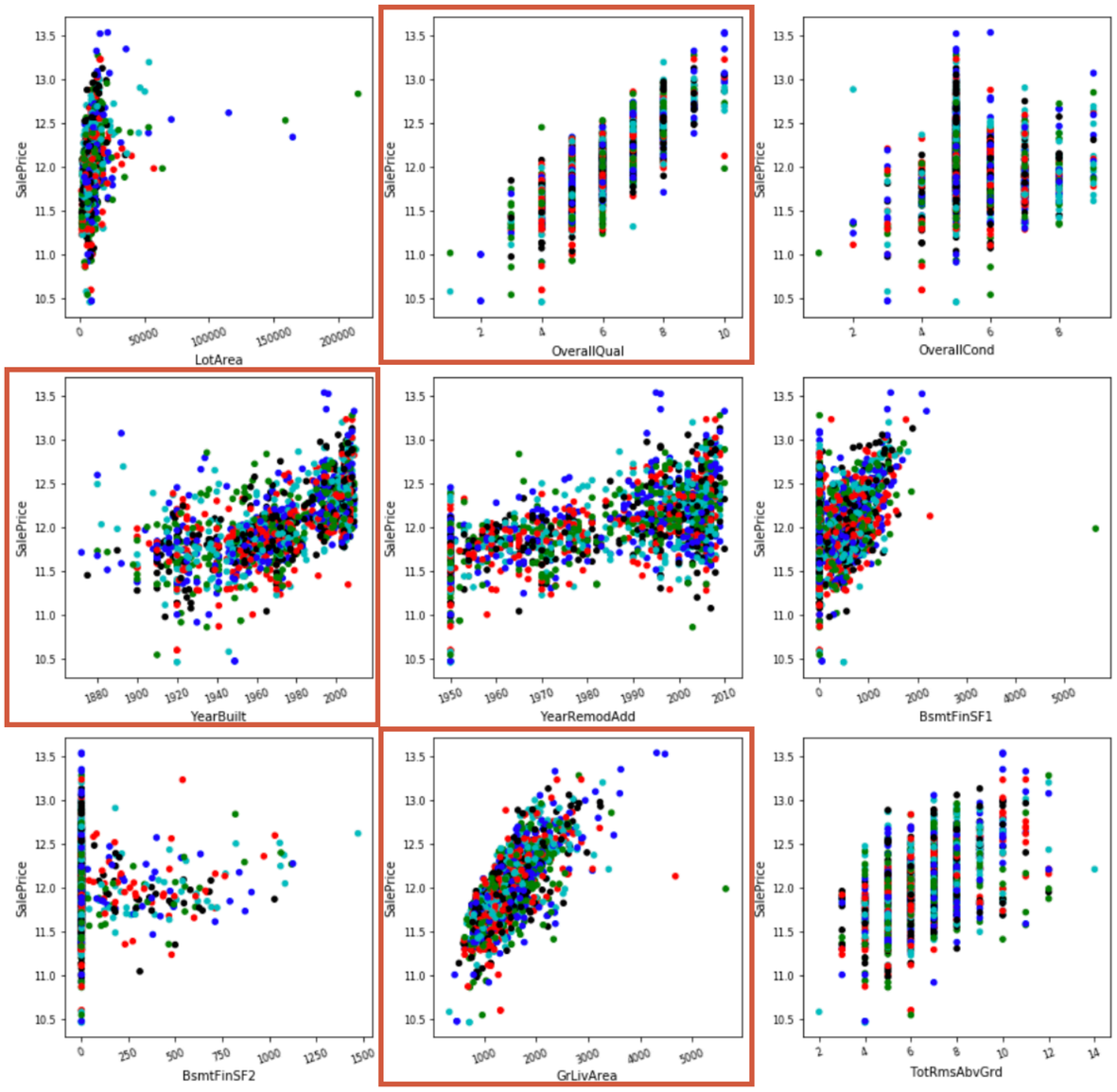
SalePrice vs Year Sold / Month Sold





Feature Engrg : Month from Jan 2016 revealed Periodic pattern for quantity of Sold Houses

Kaggle - Predicting House Prices with Advanced Regression Techniques



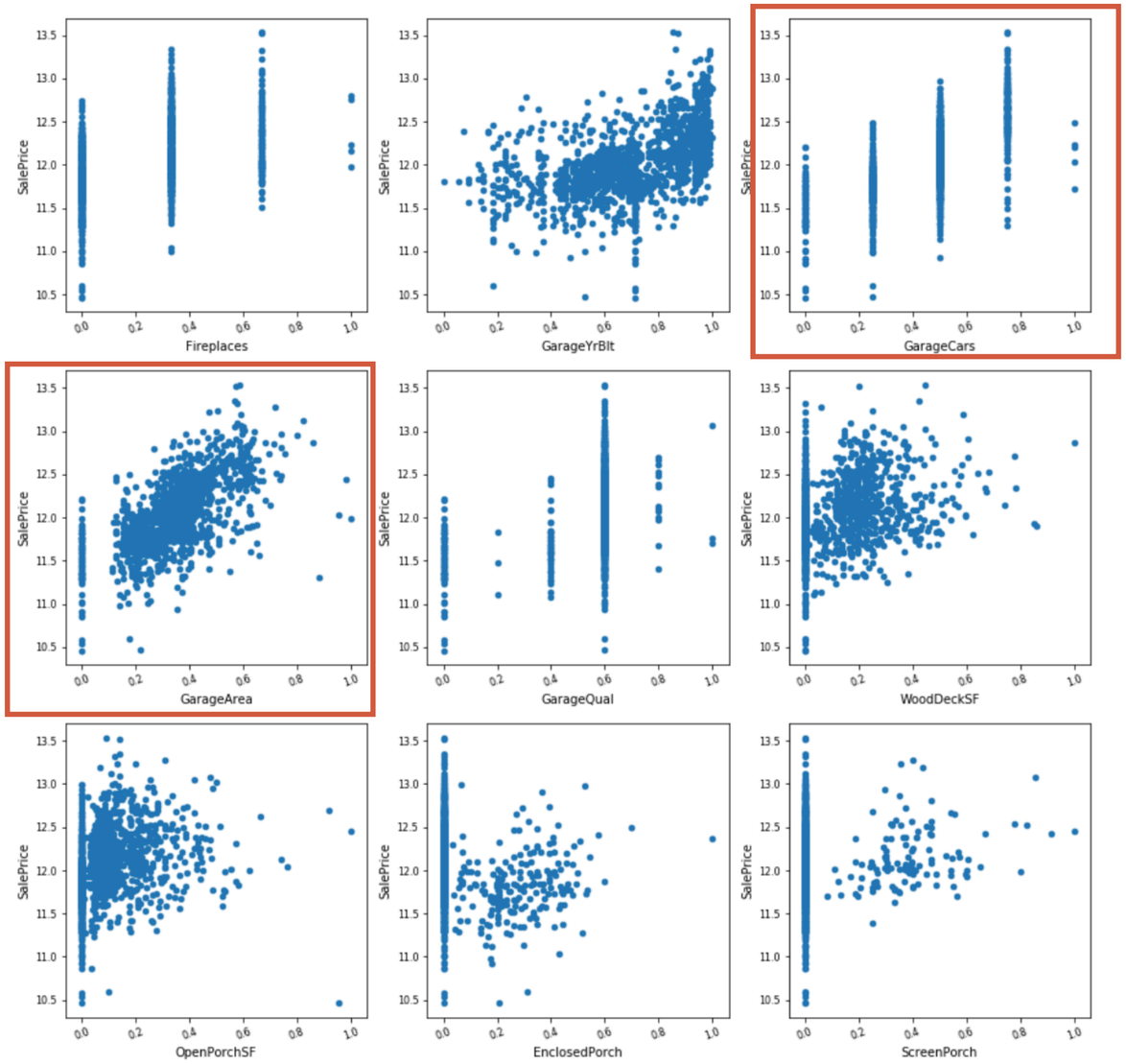
EDA

Many numeric type features seems not well correlated to Sale Price, such as ‘Year Sold’, ‘Mouth Sold’.

However, there are several variables demonstrates some extend of correlation with sale price.

These variables generally related to size of the house (area), quantity of house’s feature (no. of bedroom, garage etc) and year built or renovate of the house. Most obvious correlation was observed for ‘Overall Qual’ and ‘ GrlivAreas’.

Kaggle - Predicting House Prices with Advanced Regression Techniques



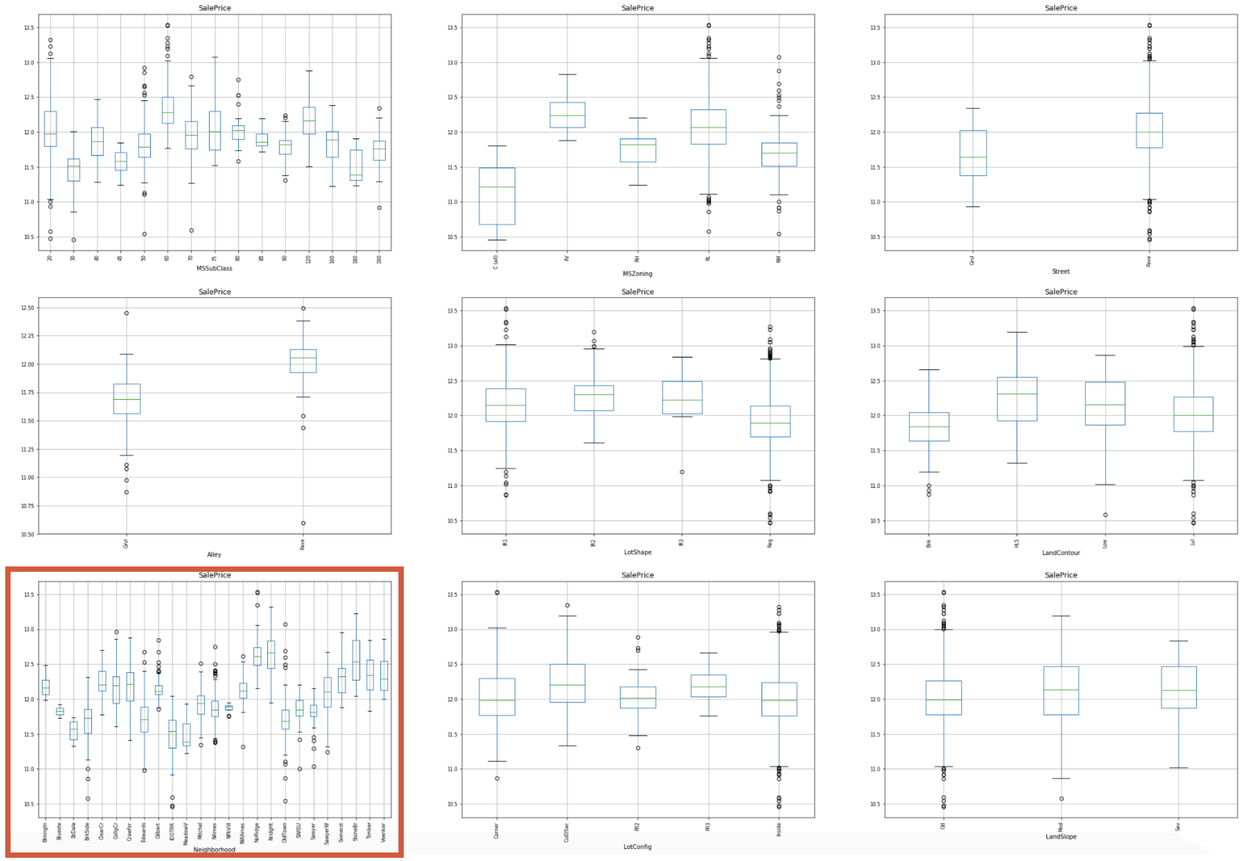
EDA

Many numeric type features seems not well correlated to Sale Price, such as ‘Year Sold’, ‘Mouth Sold’.

However, there are several variables demonstrates some extend of correlation with sale price.

These variables generally related to size of the house (area), quantity of house’s feature (no. of bedroom, garage etc) and year built or renovate of the house. Most obvious correlation was observed for ‘Overall Qual’ and ‘ GrlivAreas’.

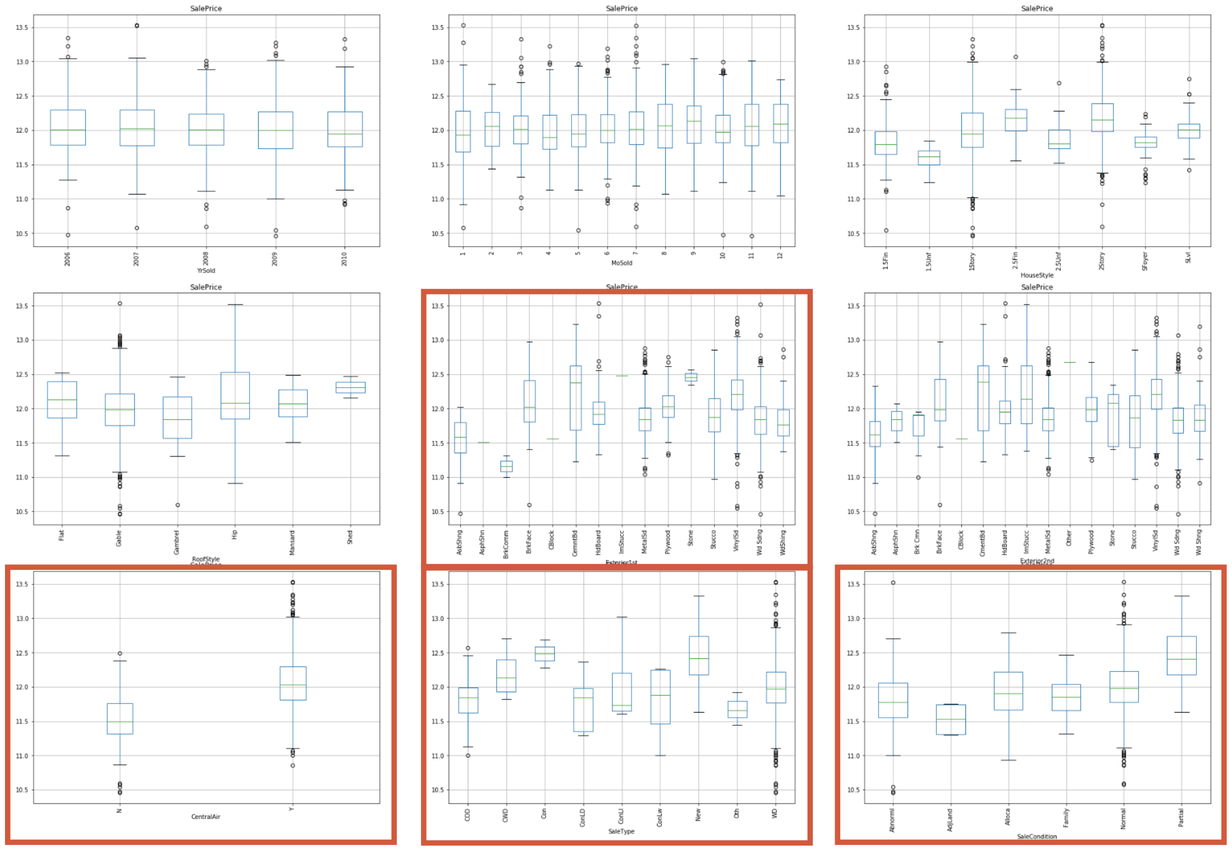
Kaggle - Predicting House Prices with Advanced Regression Techniques



EDA

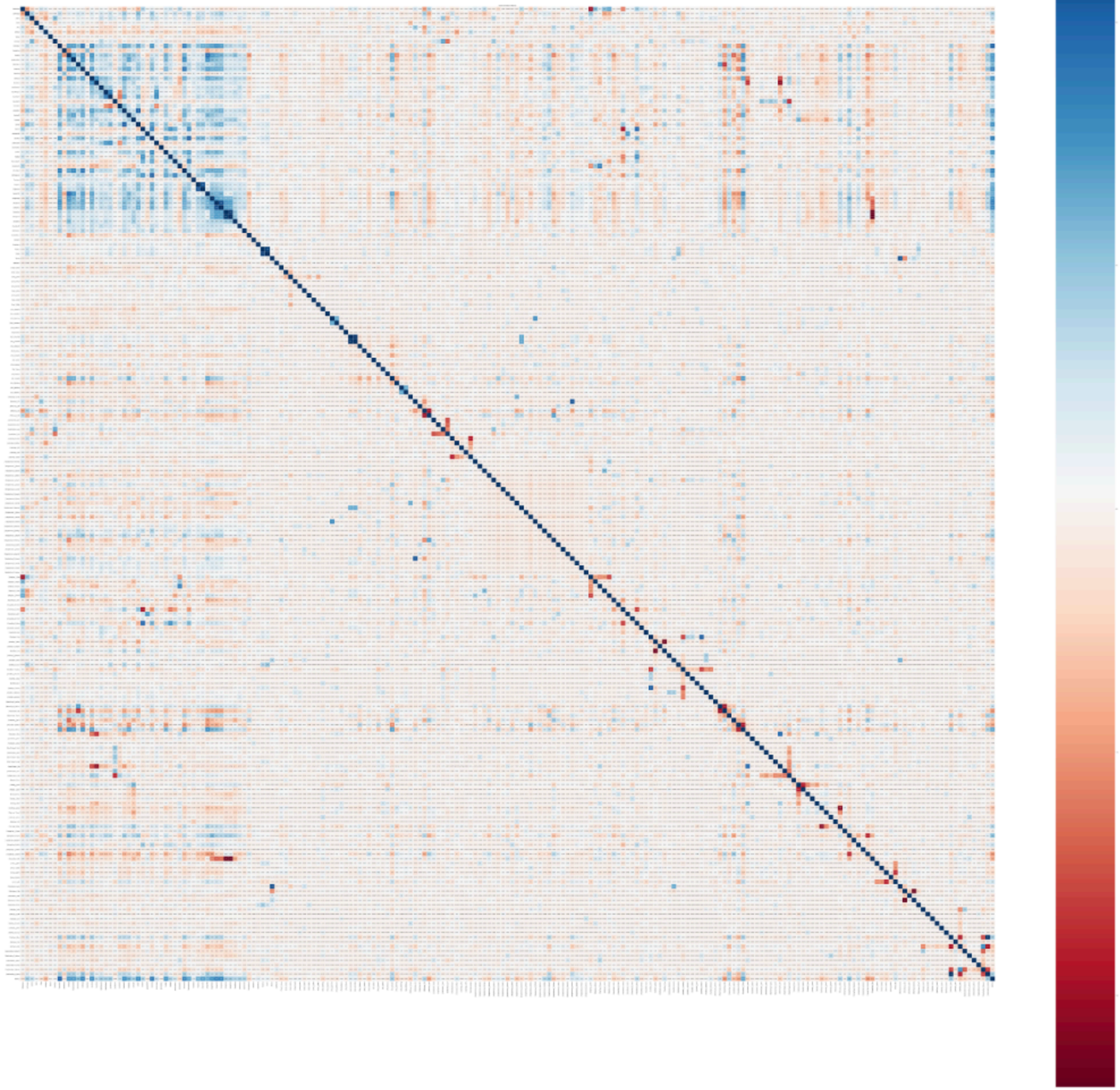
Similarly, several categorical features demonstrates some extend of variation as compared to sale price, whilst there are many don't. These features which seems impacted sale price are central air, neighbourhood, zoning, external quality, Sale Type.
Those do not vary much with price such as land slope and roof style.

Kaggle - Predicting House Prices with Advanced Regression Techniques



EDA

Similarly, several categorical features demonstrates some extend of variation as compared to sale price, whilst there are many don't. These features which seems impacted sale price are central air, neighbourhood, zoning, external quality, Sale Type.
Those do not vary much with price such as land slope and roof style.



Kaggle - Predicting House Prices with Advanced Regression Techniques

Feather Selection : Correlation

From Correlation Matrix of the features, those with deeper colour shows high correlation with other features.
These features are interdependent such as garage quality vs garage condition, garage area vs garage car.
20 of these highly correlated Features (>80%) had beed extracted and eliminated from the data.

```
Features with correlation higher than 80% : 20

Out[463]: ['1stFlrSF',
           'TotRmsAbvGrd',
           'FireplaceQu',
           'GarageArea',
           'GarageCond',
           'PoolQC',
           "['Ext_']_CmentBd",
           "['MSZoning_']_RM",
           "['Neighborhood_']_Somerst",
           "['BldgType_']_1Fam",
           "['HouseStyle_']_2Story",
           "['RoofStyle_']_Hip",
           "['RoofMatl_']_Tar&Grv",
           "['MasVnrType_']_Ne",
           "['BsmtFinType2_']_Ne",
           "['Electrical_']_SBrkr",
           "['GarageType_']_Ne",
           "['MiscFeature_']_Gar2",
           "['MiscFeature_']_Shed",
           "['SaleCondition_']_Partial"]
```

Kaggle - Predicting House Prices with Advanced Regression Techniques

Lasso

RMSE Results

```
In [247]: alpha = None
reg = LassoCV(eps=0.001, n_alphas=100,cv=5)
reg.fit(X_train, y)

cv_mse_mean = np.mean(reg.mse_path_, axis=0)
cv_rmse_mean = np.sqrt(cv_mse_mean)
cv_mse_var = np.var(reg.mse_path_, axis=0)
Best_Alpha = reg.alpha_
print("alphas: %s" % alpha)
print("CV RMSE: %s" % cv_rmse_mean)
print("CV MSE VAR %s" % cv_mse_var )
print("Best alpha using built-in LassoCV: %f" % Best_Alpha)
executed in 241ms, finished 21:58:17 2019-04-29

alphas: None
CV RMSE: [0.19692041 0.23123213 0.21628923 0.19534973 0.21872052]
CV MSE VAR [0.00152978 0.00209756 0.00184322 0.00137159 0.00127195]
Best alpha using built-in LassoCV: 0.000396
```

RFECV

RMSE Results

```
In [251]: model = LinearRegression()
x, y =X_train, y

rfecv_S = RFECV(model, cv=5, step=1, scoring="neg_mean_squared_error", min_features_to_select=90)
rfecv_S.fit(x, y)

rfecv_S.transform(x)
print(rfecv_S.n_features_)
print(rfecv_S.transform(x))
print(rfecv_S.ranking_)
executed in 5.86s, finished 21:59:17 2019-04-29

105
[[0.23529412 0.15068493 0.0334198  ... 0.         0.         1.        ]
 [0.         0.20205479 0.03879502 ... 0.         0.         1.        ]
 [0.23529412 0.1609589  0.04650728 ... 0.         0.         1.        ]
 ...
 [0.29411765 0.15410959 0.03618687 ... 0.         0.         1.        ]
 [0.         0.1609589  0.03934189 ... 0.         0.         1.        ]
 [0.         0.18493151 0.04037019 ... 0.         0.         1.        ]]
[ 1  1  1  1 47 85  1 81  1  1  1 23 82 67 40  1 54  1 38  1  1  1  1  1
  1  8 48  1  1 72  2  4 11  1  1  1  1 74 43  1  1 34  1  1  1  1  1  1
 75  1  9  1 59  1  1 53  1  1 51  1 25  1  1  1 55 77 58 41  1 80  7 26
 46 84 31 57  1  1  1  1  1  1  1  1  1  1  1  1  1 44 13  5 49 71 19  1
  1 18 12  1 10 15 24 14 70  1  6 20 16 17  1 45 69  1 68  1 65 79 60 21
  1  1 64 42 37 73 61 32  1  1 33  1  1 52 86  1  3 78 39  1  1  1  1  1
  1 62 27 83 29 28 30  1  1  1  1  1  1 50 63 56 22  1  1  1  1  1  1  1
  1  1  1  1 76  1  1  1  1  1  1  1  1  1  1 36  1  1 35  1 66]
```

```
In [252]: f = rfecv_S.get_support(1)                    #the most important features
rfecv_L = X_train[X_train.columns[f]]        # the most important features
# rfecv_L.columns
# rfecv_L.shape
print(rfecv_L.shape)
executed in 8ms, finished 21:59:23 2019-04-29

(1460, 105)
```

```
In [253]: cv_grid_rmse = np.sqrt(-rfecv_S.grid_scores_)
cv_grid_rmse.mean()
executed in 7ms, finished 21:59:25 2019-04-29

Out[253]: 10120872337.498697
```

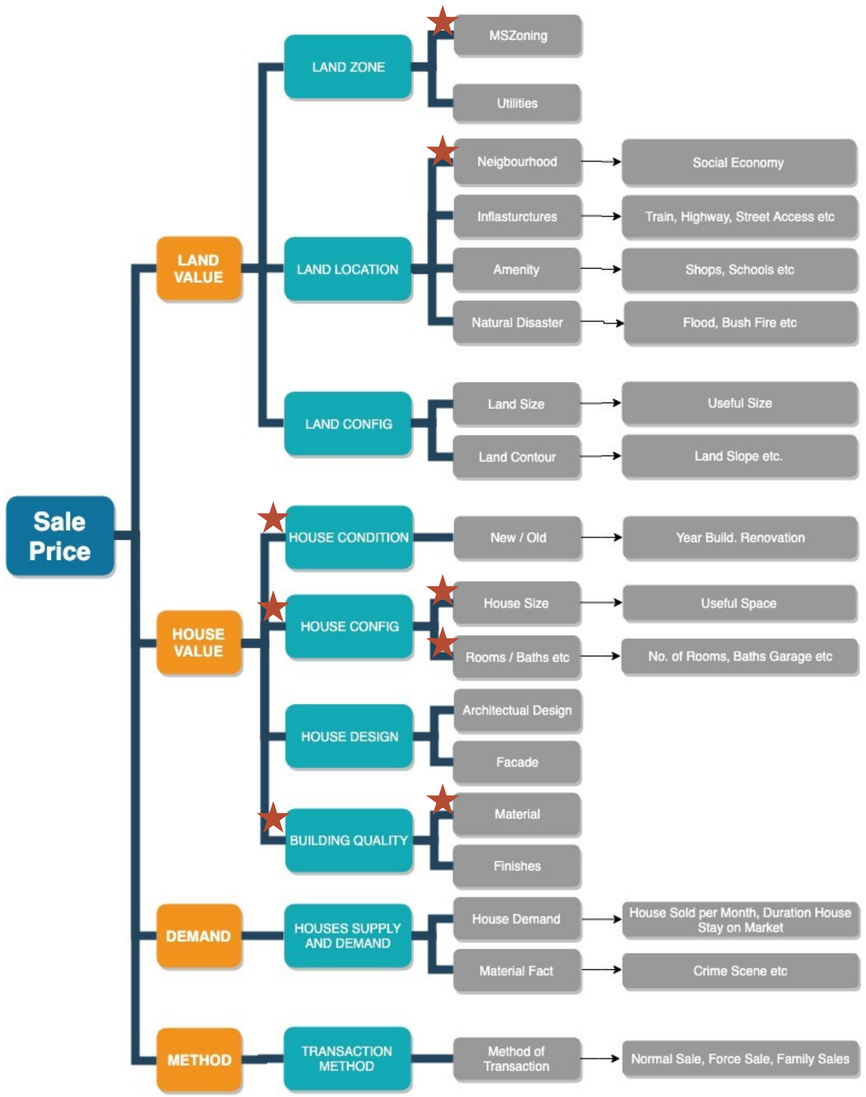
```
In [254]: rfecv_S.score(x,y)
executed in 13ms, finished 21:59:28 2019-04-29

Out[254]: 0.9283232173837994
```

Feather Selection : Lasso Vs RFECV

Evaluation have been carried out with cross validation of N = 5 for both Lasso Regression and RFECV Regression.
Lasso (alpha 0.000396) picked 101 variables and eliminated the other 89 variables.
RFECV (min feature = 90) picked 105 important features

After comparing the cross validation for both Lasso and RMSE, I decided to pick features selected Lasso Regression as it has more reasonable RMSE mean and variance.

Kaggle - Predicting House Prices with Advanced Regression Techniques



Features Selected by Lasso
Top 20 and Bottom 20

	0
GrLivArea	1.263102
['RoofMatl_']_ClyTile	1.214130
OverallQual	0.500579
['MSZoning_']_C (all)	0.279238
OverallCond	0.249728
GarageCars	0.224377
TotalBsmtSF	0.192909
Functional	0.157286
['Neighborhood_']_StoneBr	0.125894
BsmtFullBath	0.111187
['Neighborhood_']_NridgHt	0.107273
['Neighborhood_']_Crawfor	0.107043
YearBuilt	0.106785
Fireplaces	0.098552
ScreenPorch	0.096293
['MSZoning_']_FV	0.093660
FullBath	0.085816
LotArea	0.083771
['BldgType_']_Twnhs	0.080993
BsmtQual	0.080403

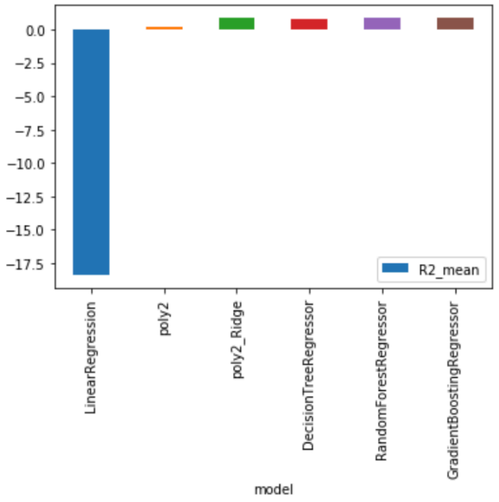
	0
['LandContour_']_Lvl	0.000085
['MasVnrType_']_BrkFace	0.001172
['GarageType_']_Detchd	0.001710
['Ext_']_Plywood	0.002003
['LandContour_']_Low	0.002278
['MiscFeature_']_Ne	0.002309
LotShape	0.002498
['LotConfig_']_FR2	0.004622
['Neighborhood_']_Timber	0.004740
['Fence_']_Ne	0.005351
LandSlope	0.005544
['Neighborhood_']_NAmes	0.005582
['Ext_']_VinylSd	0.005891
['LandContour_']_HLS	0.006762
['BsmtFinType2_']_ALQ	0.007139
ExterQual	0.007192
['SaleCondition_']_Family	0.007235
['SaleType_']_ConLD	0.007838
['LotConfig_']_Corner	0.008473
['Neighborhood_']_NWAmes	0.008525

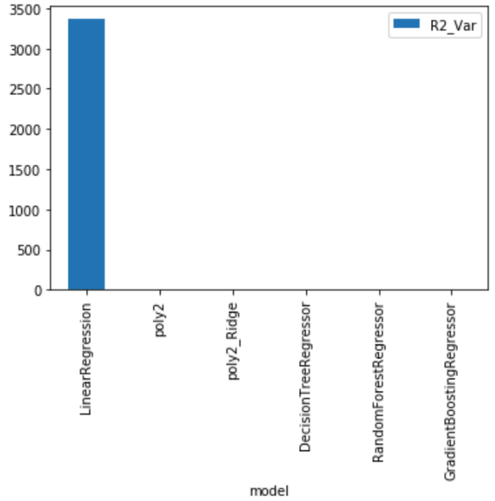
Feather Selection : Lasso Vs RFECV

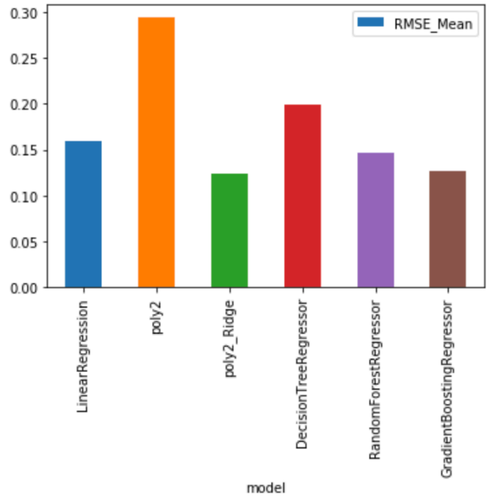
Evaluation have been carried out with cross validation of N = 5 for both Lasso Regression and RFECV Regression.
Lasso (alpha 0.000396) picked 101 variables and eliminated the other 89 variables.
RFECV (min feature = 90) picked 105 important features

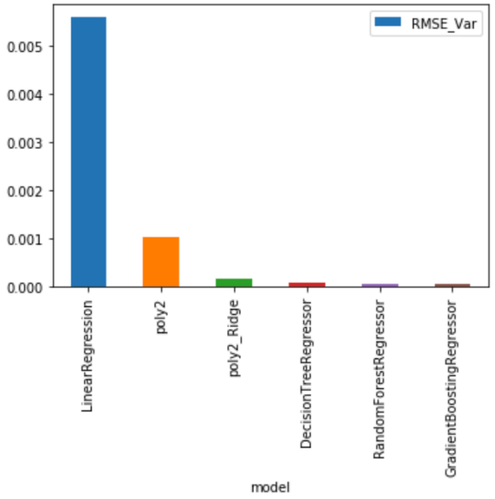
After comparing the cross validation for both Lasso and RMSE, I decided to pick features selected Lasso Regression as it has more reasonable RMSE mean and variance.

model	R2_mean	R2_Var	RMSE_Mean	RMSE_Var
LinearRegression	-18.429495	3364.869198	0.159752	0.005603
poly2	0.241646	0.018414	0.293970	0.001029
poly2_Ridge	0.902511	0.001032	0.124070	0.000160
DecisionTreeRegressor	0.739740	0.001387	0.198654	0.000079
RandomForestRegressor	0.864802	0.000783	0.147563	0.000108
GradientBoostingRegressor	0.902460	0.000502	0.125908	0.000064









Kaggle - Predicting House Prices with Advanced Regression Techniques

Modeling and Evaluation

With 79 features, this dataset has quite low number of instances, i.e. 1460. I decided to use cross validation of 10 to build and verify the Models.

Model

Base Model : Linear Regression (LR)

Evaluate Models:

1. Polynomial (order 2) regression (Poly2R)
2. Polynomial (order 2) + Ridge Regression (Poly_RidgeR)
3. Decision Tree Regression (DTR)
4. Random Forest Regressing (RFR)
5. Gradient Boosting Regression (GBR)

Summary

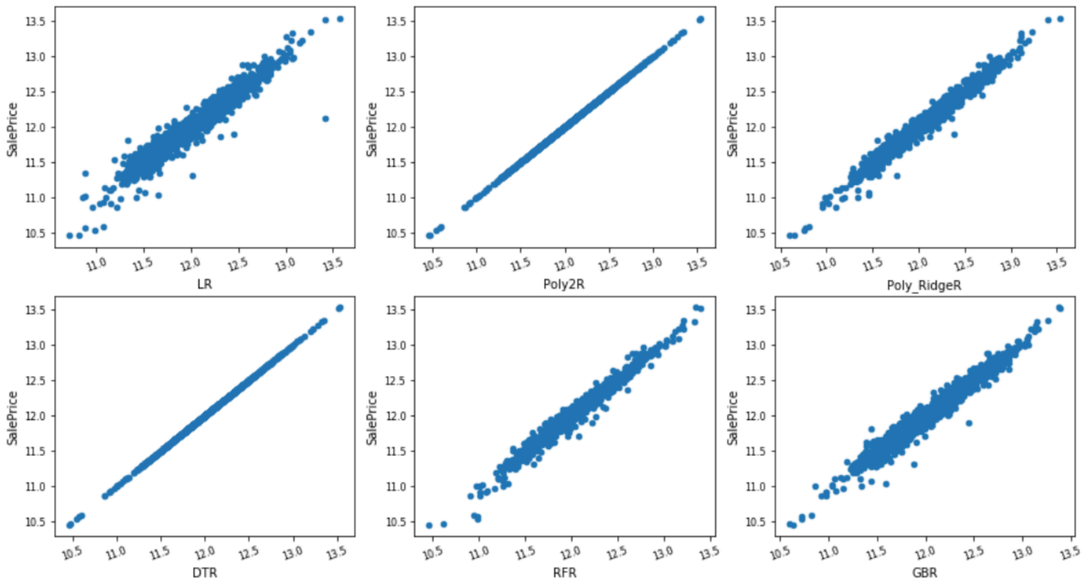
Base Model : High variance and very bad prediction

Evaluate Models:

1. GBR has the best combination of Mean and variance performances for both R2 and RMSE. R2(0.90) and RMSE (0.126)
2. Poly_RidgeR had similar performances as as GBR, except with variance slightly over double of GBR
3. It is noticed tha Sale Price Predict is same as Sale Price (Target) for Poly2R and DTR. It seems that these 2 models are completely overfitted, Will need to investigate further.

Kaggle - Predicting House Prices with Advanced Regression Techniques

Sale Price Vs Sale Price Predict



Modeling and Evaluation

With 79 features, this dataset has quite low number of instances, i.e. 1460. I decided to use cross validation of 10 to build and verify the Models.

Model

Base Model : Linear Regression (LR)

Evaluate Models:

1. Polynomial (order 2) regression (Poly2R)
2. Polynomial (order 2) + Ridge Regression (Poly_RidgeR)
3. Decision Tree Regression (DTR)
4. Random Forest Regressing (RFR)
5. Gradient Boosting Regression (GBR)

Summary

Base Model : High variance and very bad prediction

Evaluate Models:

1. GBR has the best combination of Mean and variance performances for both R2 and RMSE. R2(0.90) and RMSE (0.126)
2. Poly_RidgeR had similar performances as as GBR, except with variance slightly over double of GBR
3. It is noticed tha Sale Price Predict is same as Sale Price (Target) for Poly2R and DTR. It seems that these 2 models are completely overfitted, Will need to investigate further.

WHAT'S NEXT?

Kaggle - Predicting House Prices with Advanced Regression Techniques

Further Improvement

Due to time constraint, there are several steps in this project which can be revisit in future to improve the outcome.

1. Refine the data pre-processing : impute with interpolation instead on taking the mean value (numeric), mode (categorical)
2. Refine EDA and feature Engineering : there are room to investigate to engineer some of the features such as size related features
3. Refine the Hyperparameters : to refine using GridSearch in state of manual tuning for Lasso, Ridge and other models