CS 4342 Term Project
Group member: Yueting Zhu, George Chakhnashvili, Ivan Martinovic

Feature engineering:

1. Data modification
1. PCA
2. Feature selection

ML Algorithm choices:

1. Linear (Regression)
2. Multi feature (Regression)
3. Lasso/Ridge (Regression)
4. Logistic Regression (Classification)
5. LDA/QDA (Classification)
6. Trees (Regression)
7. SVC (Classification)

For this project, our team will be using a data set about houses found in Kaggle.
Link: House Prices - Advanced Regression Techniques | Kaggle

Learning objective:

Our goal is to predict the price of the house with other given features, such as living size, garden size, and fireplace size. (Regression)

Predict whether the price of a specific house will be higher or lower than the median house price with other given features, such as living size, garden size, and fireplace size. (Classification)

Description of the dataset:
Shape: (1460,81)

Original file is around 460 kB in csv format. As readers can tell, the size of the data will be relatively big, so we expect that feature engineering will take us most of the time.

Variables:

- 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

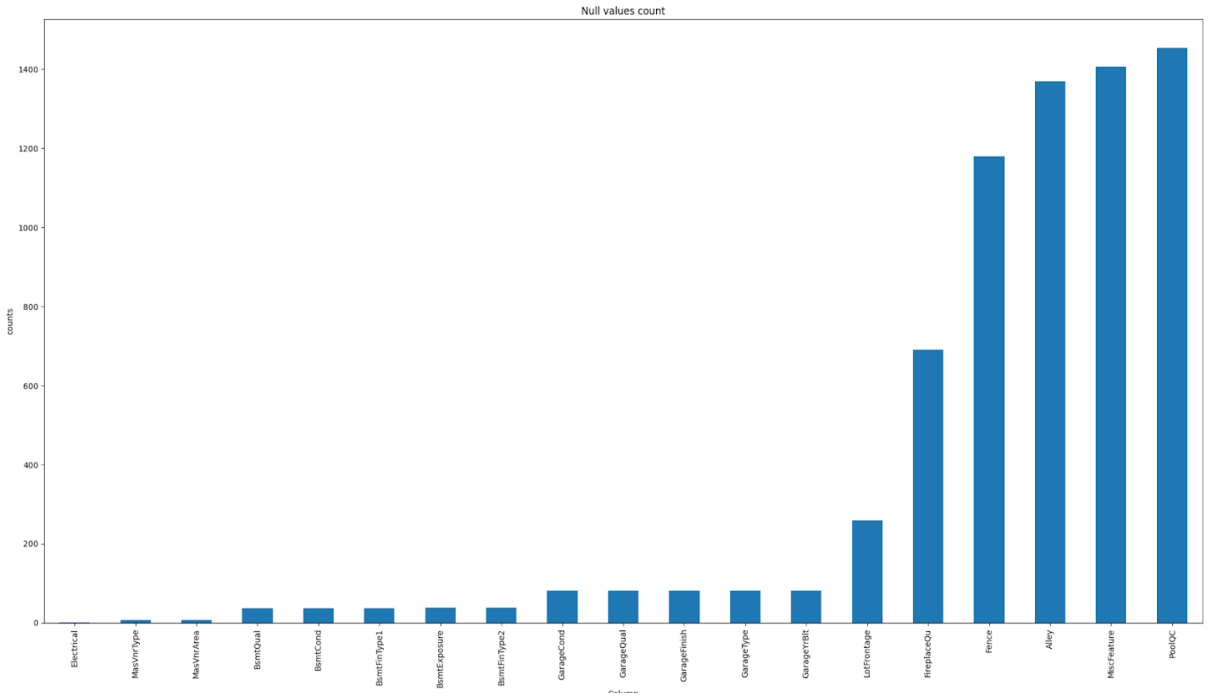
- Utilities: Type of utilities available
- LotConfig: Lot configuration
- LandSlope: Slope of property
- Neighborhood: Physical locations within Ames city limits
- Condition1: Proximity to main road or railroad
- Condition2: Proximity to main road or railroad (if a second is present)
- BldgType: Type 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

In order to visualize this dataset, we will be conducting a PCA and feature selection on the dataset.

To clean up all categorical variables, we turned all of them into dummy variable columns. At the end, we have a much bigger data size of (1460,239).

However, we have noticed that some columns have some missing values, meaning that there are lots of data missing. For example, LotFrontage which is numeric had some NA values.



Column names vs. value missing

So we decided to drop all rows with at least one value missing and we still get a relatively big data size of (585,239).

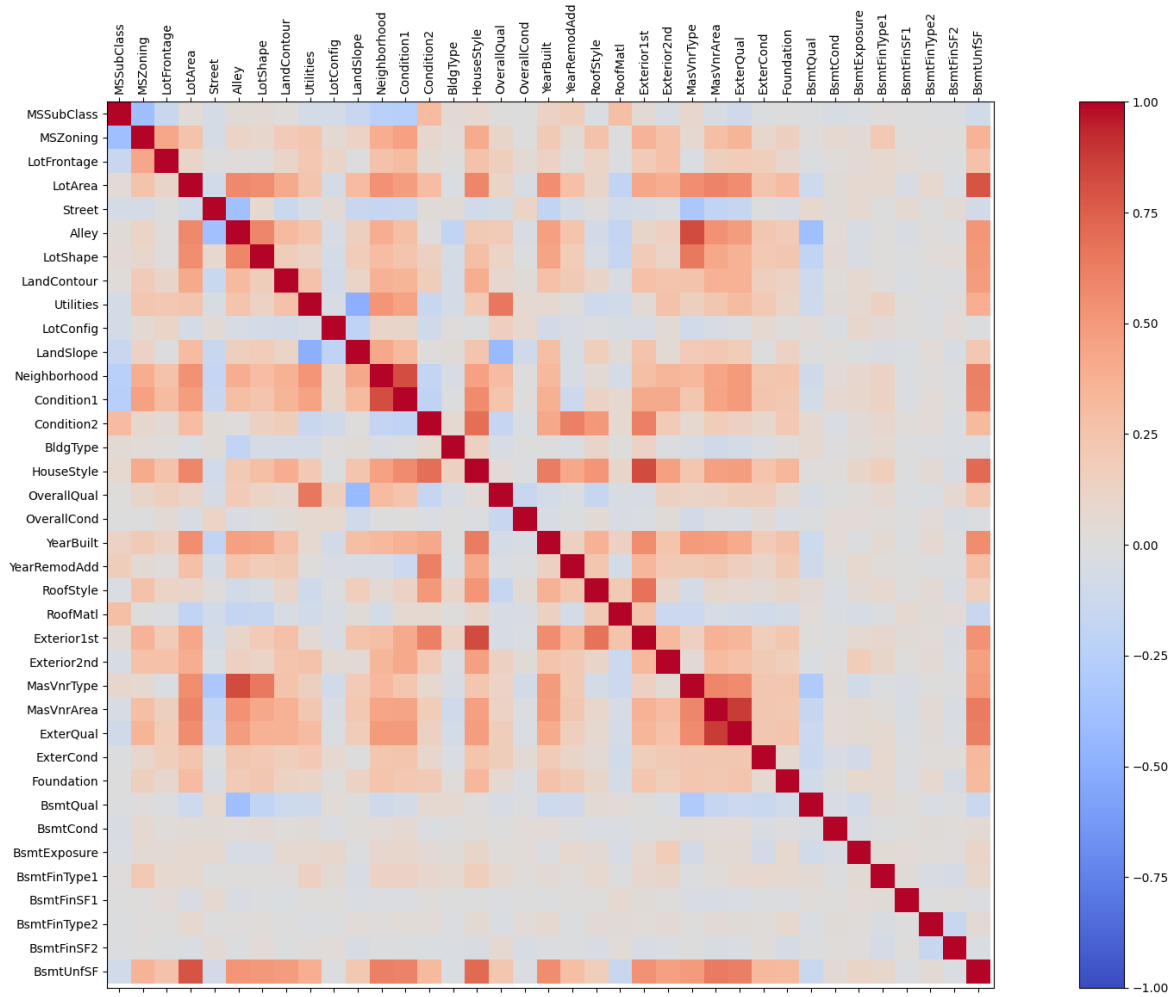
For a glance of the data, we printed out the head of the data

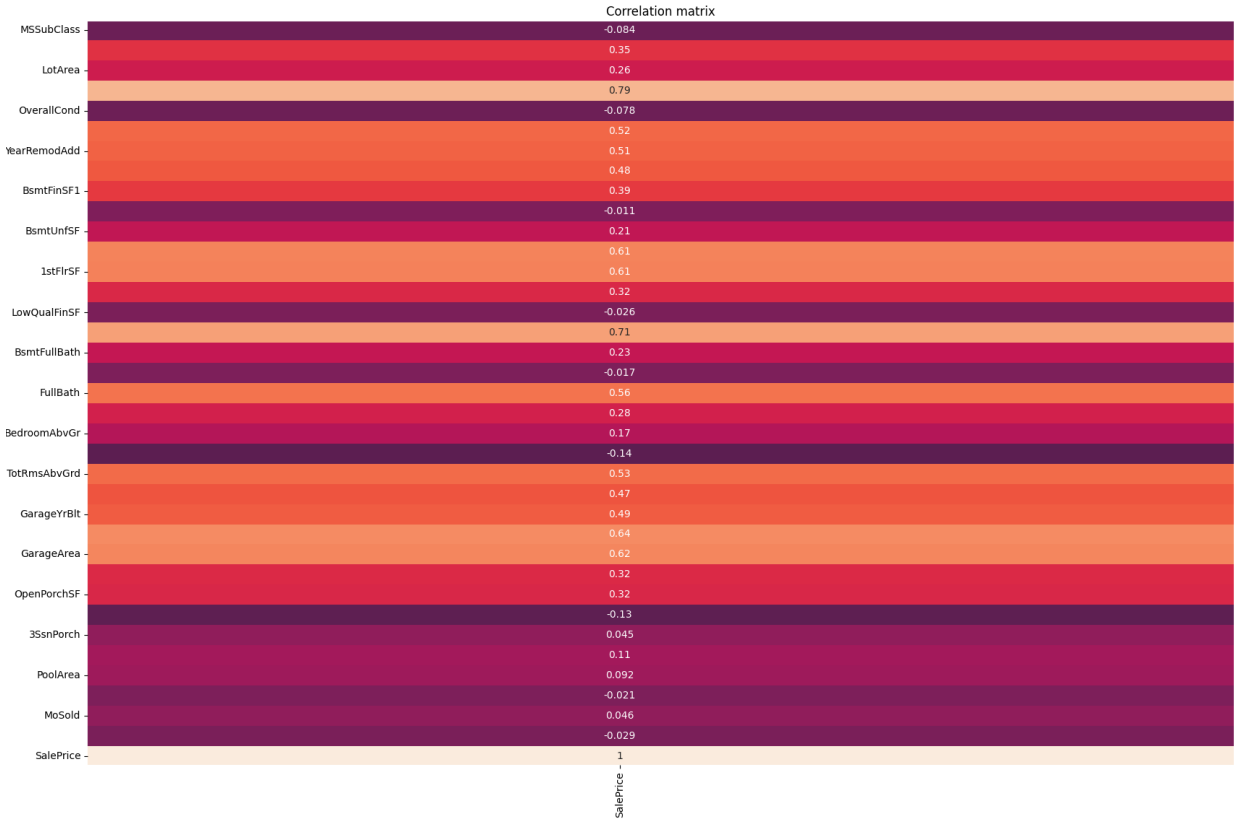
	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF
0	80.0	9600	6	8	1976	1976	0.0	978	0	284	1262
1	68.0	11250	7	5	2001	2002	162.0	486	0	434	920
2	60.0	9550	7	5	1915	1970	0.0	216	0	540	756
3	84.0	14260	8	5	2000	2000	350.0	655	0	490	1145
4	75.0	10084	8	5	2004	2005	186.0	1369	0	317	1686

5 rows × 238 columns

And there are many rows as you scroll to the right.

We also created the heatmap:

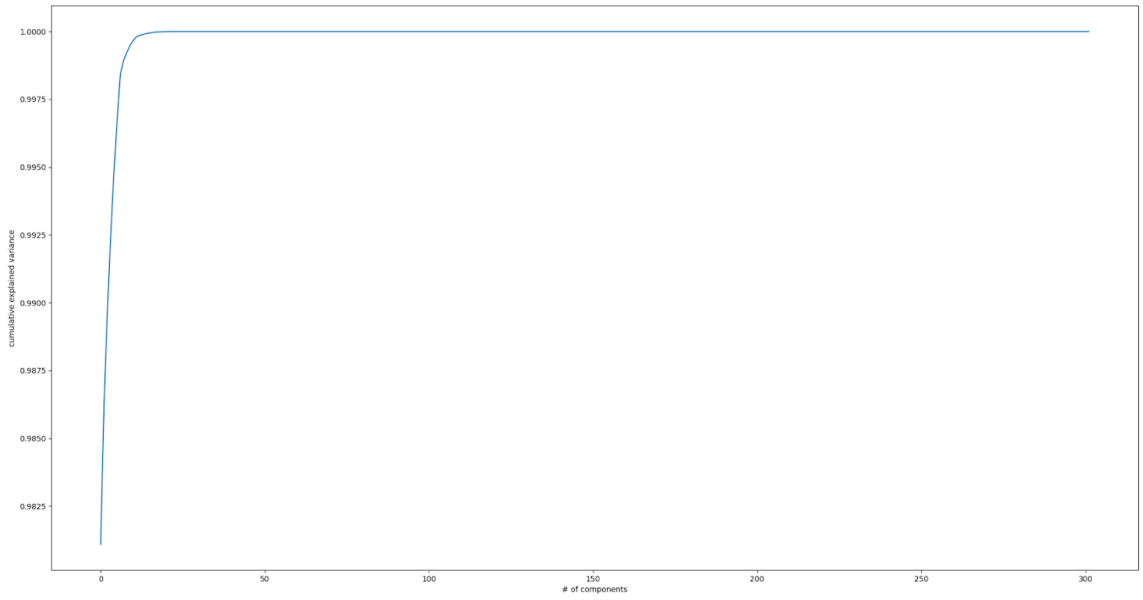

MSSubClass
MSZoning
LotFrontage
LotArea
Street
Alley
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
MasVnrType
MasVnrArea
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinSF1
BsmtFinType2
BsmtFinSF2
BsmtUnfSF
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00



Correlation matrix, SalePrice column

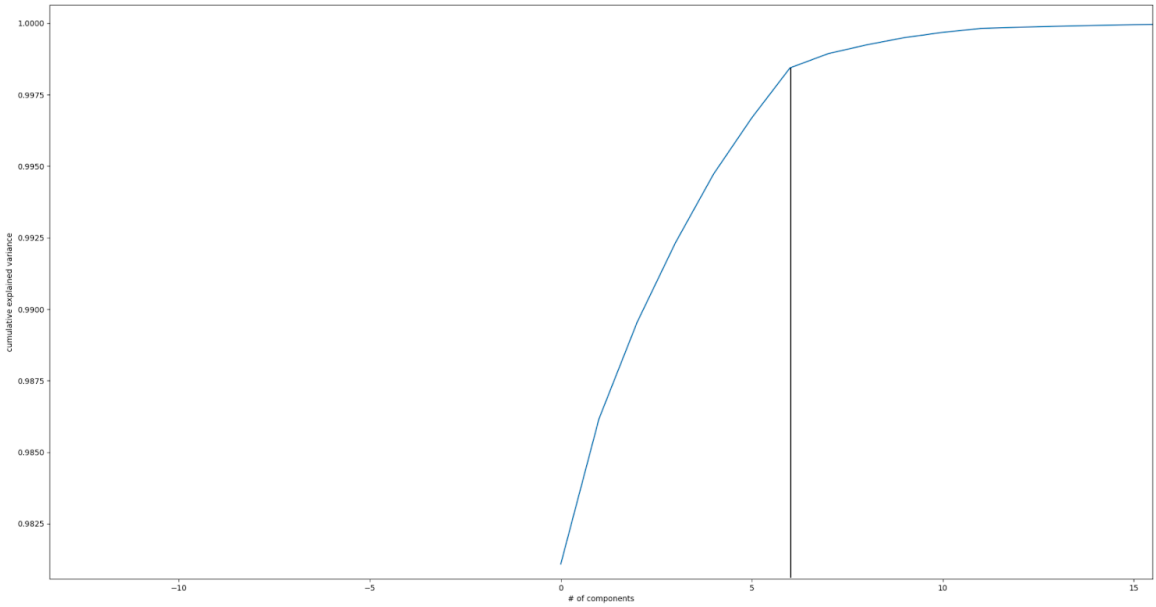
PCA:

Explained variance vs. component counts:



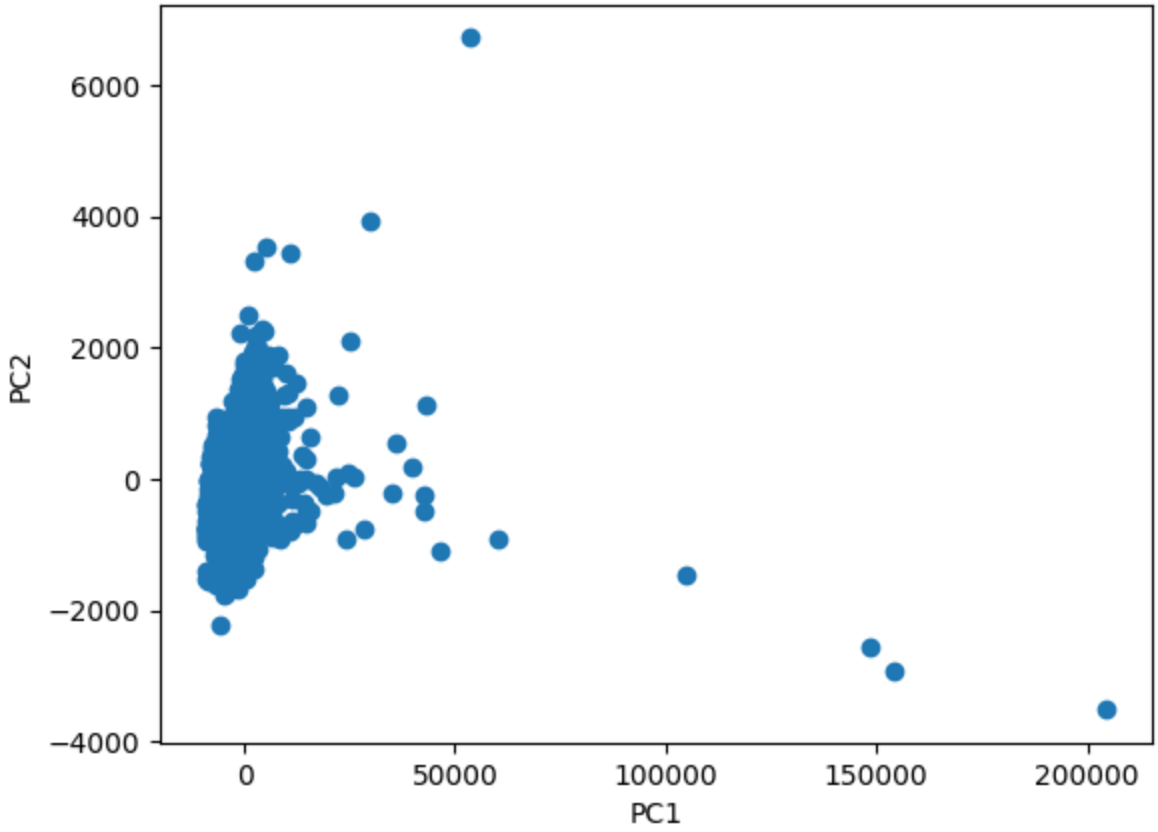
As readers can see, the number of components is very large, but the graph is starting at around 98%, so the PCA method is a very good method to visualize the dataset

In particular, we were interested in finding the optimal number of components.



If we take a closer look at the graph, we can see that after component number 6, we have a significant drop in the overall increase in explained variance. Therefore, component number 6 is the best choice if we are using PCA.

To visualize the dataset, we still decided to do a simple and easy to understand PCA graph with only 2 components, and this PCA also explain around 98% of the whole dataset according to the calculation:



PC1 Vs. PC2 visualization of the dataset

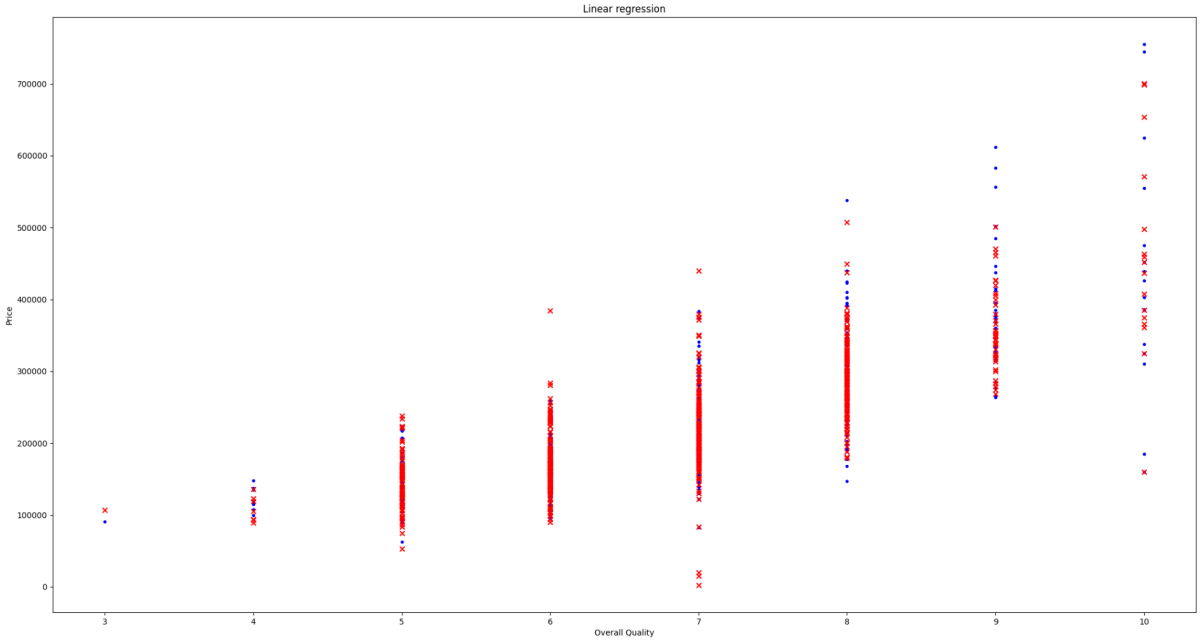
As we can see, the data points follow a nonlinear pattern, so we expect that the complex models are going to work better than the simple model like linear regression.

Prediction process:

A - Predicting value of Sale Price

1. Linear Regression

We used all predictors for Linear regression, and we plotted a graph of prediction (red with shape x), and the actual values (blue with shape dot) against the Overall quality of the house to visualize it. The reason for choosing the overall quality is that shown in the correlation matrix, this predictor is having a correlation of 0.79 with the price, so we think this is the graph that can best represent the prediction.

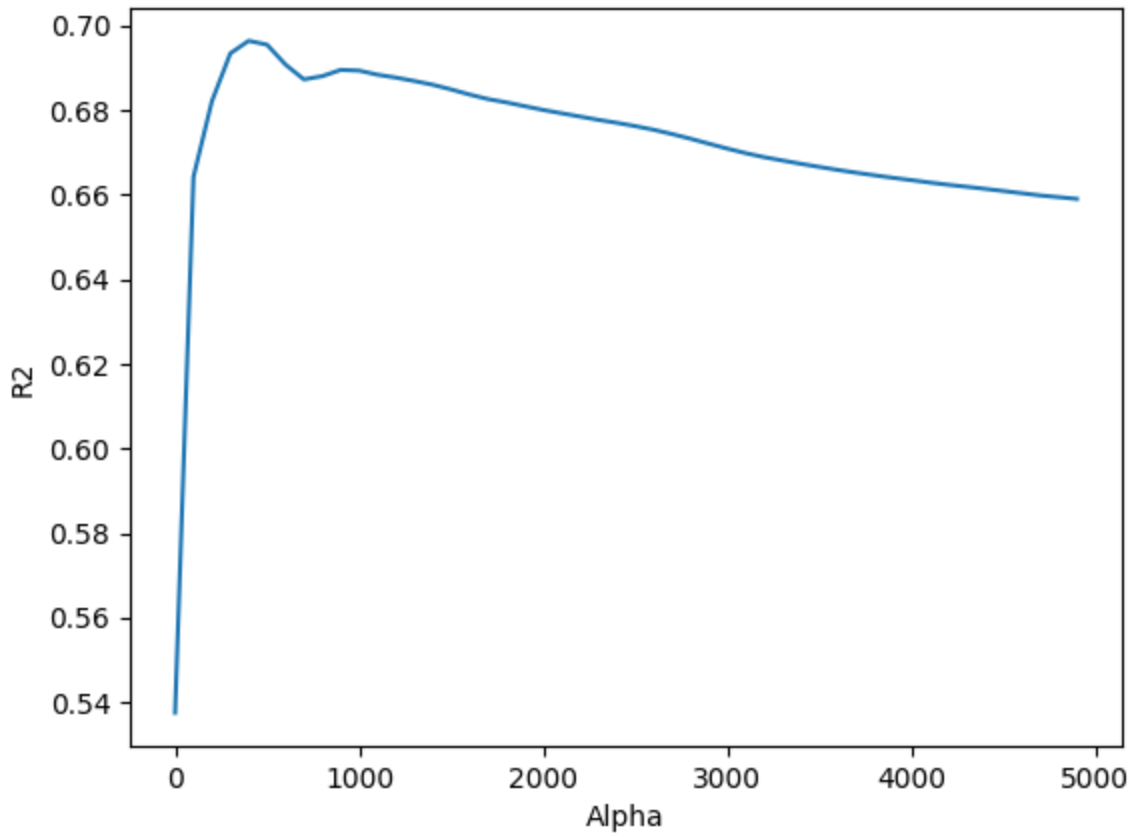


Negative mean absolute error: -34699.21373663926
max error: -439514.5865135303
explained variance: 0.5377427572840439
R2: 0.5293985186292258

The R2 is not looking good with only 50%, and the max error rate is significant, too. Overall, the prediction is not accurate enough to predict the price of the house.

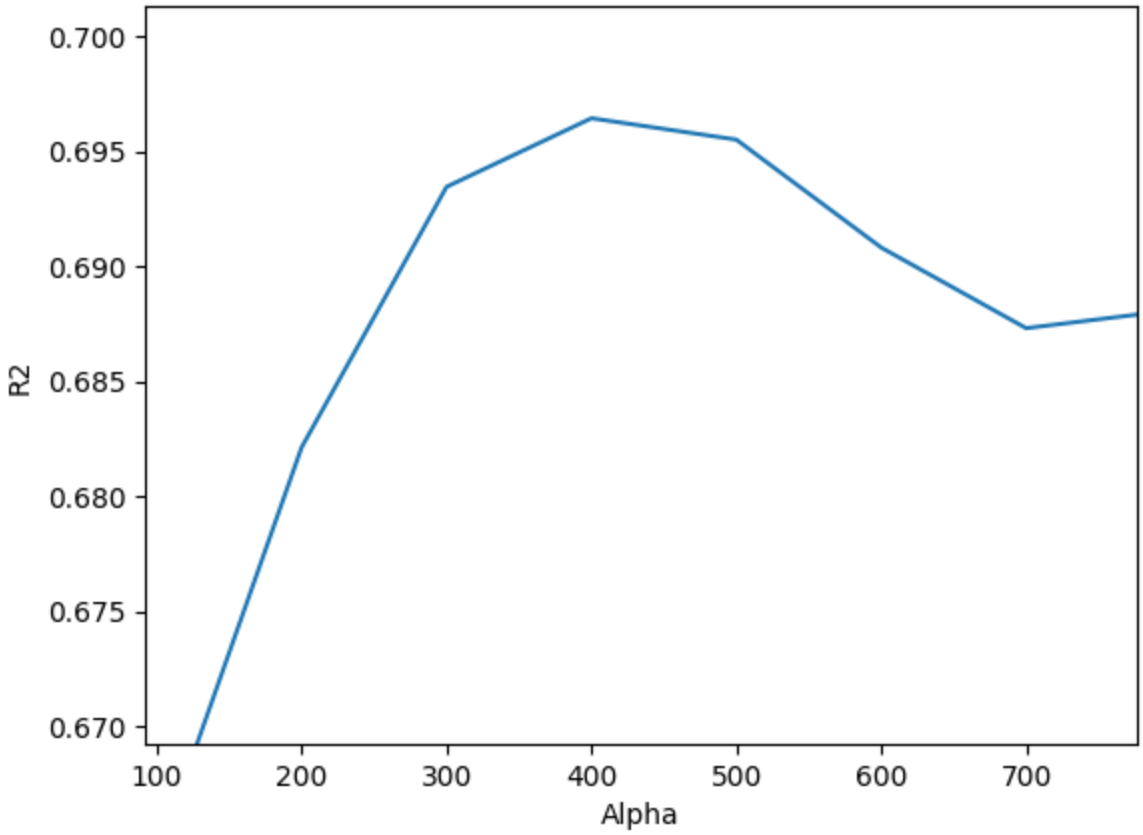
2. Lasso

To determine Alpha, we plotted the explained alpha against each alpha:



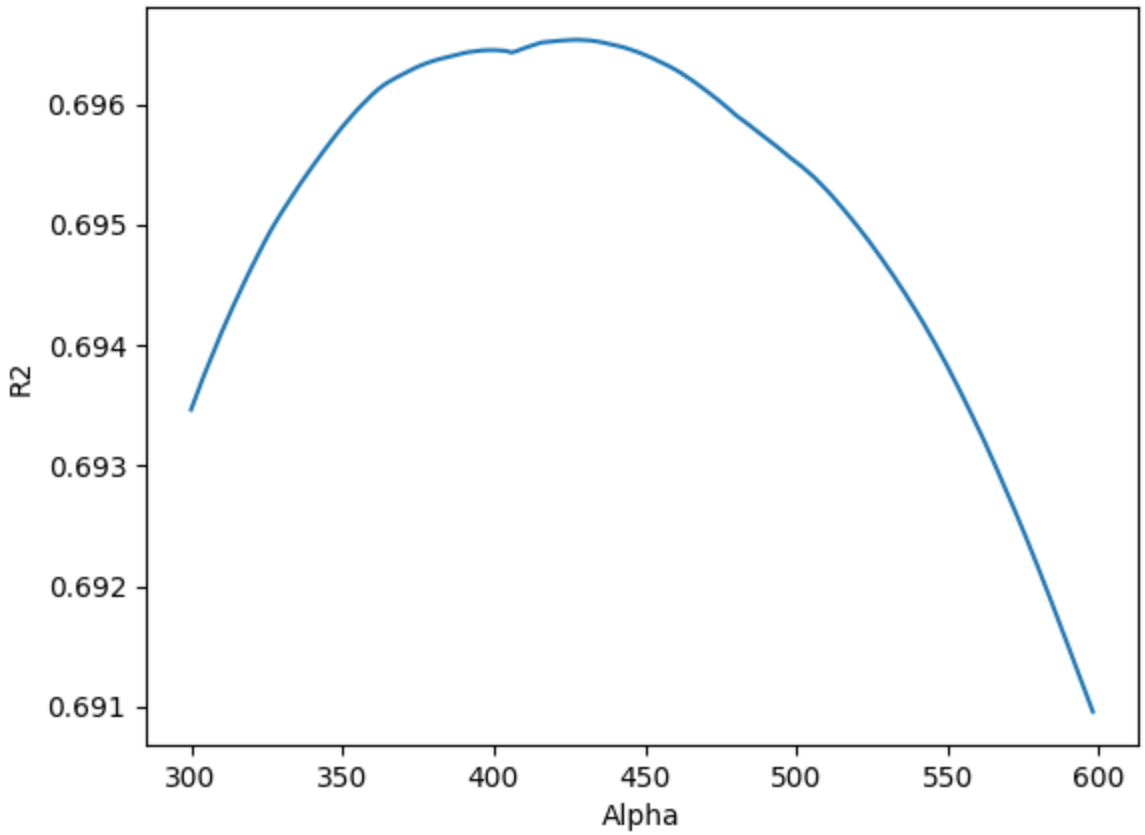
Lasso Alpha vs. R2

The alpha in this case is on a very large scale, and the alpha that is providing the best output is between 300 and 500 if we zoom in.



Zoomed in Alpha vs. R2

So we decided to try out more Alphas between 300 and 600.

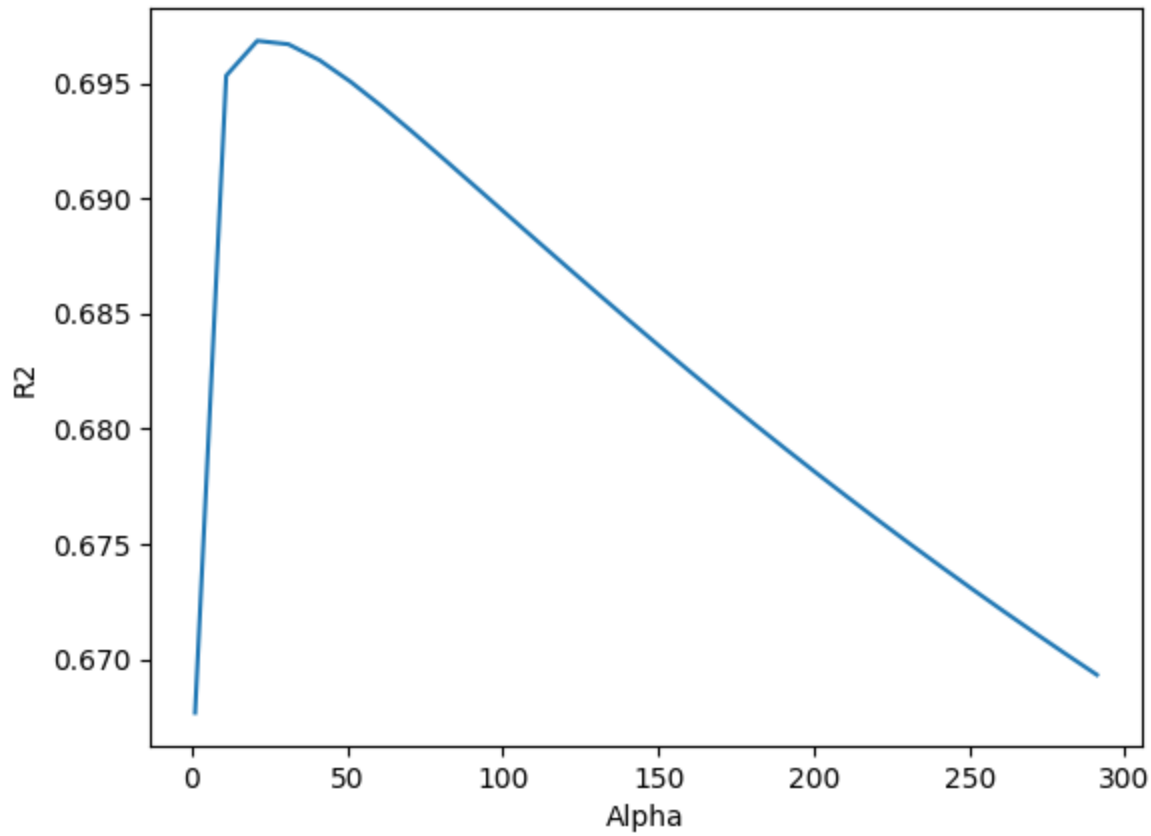


Closer steps with Alpha range 300 to 600

Now, we have a much smoother curve when trying out more Alphas between 300 and 600.
According to computer,
Best alpha is 428
Highest R2 with this alpha is 0.6965373386228715

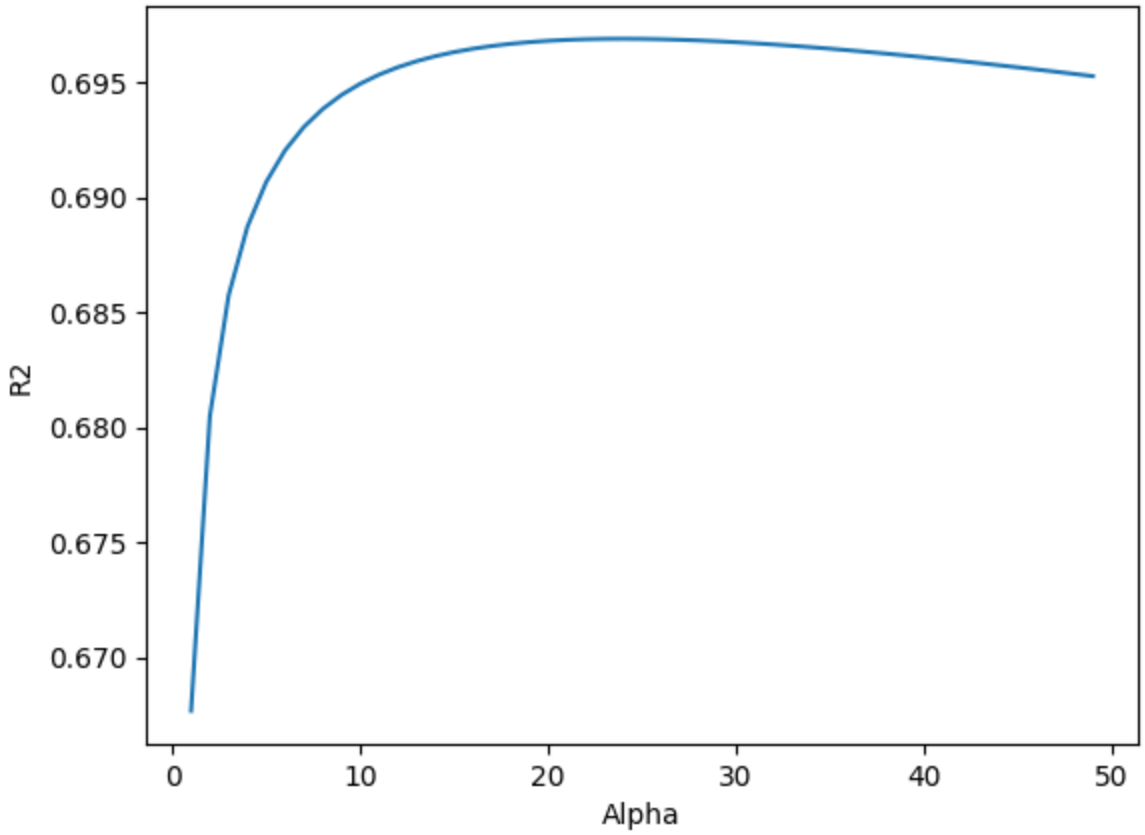
3. Ridge

For Ridge regression, we performed the same method used in Lasso, which is finding the best Alpha and then determining if the model is accurate or not.
To determine Alpha, we plotted the explained alpha against each alpha:



Ridge regression Alpha Vs. R2

As we tested Alpha in steps of 10s from 0 to 300, we found that the maximum R2 is around 70% when Alpha equals to 30, so we took a closer look at that part of the graph.

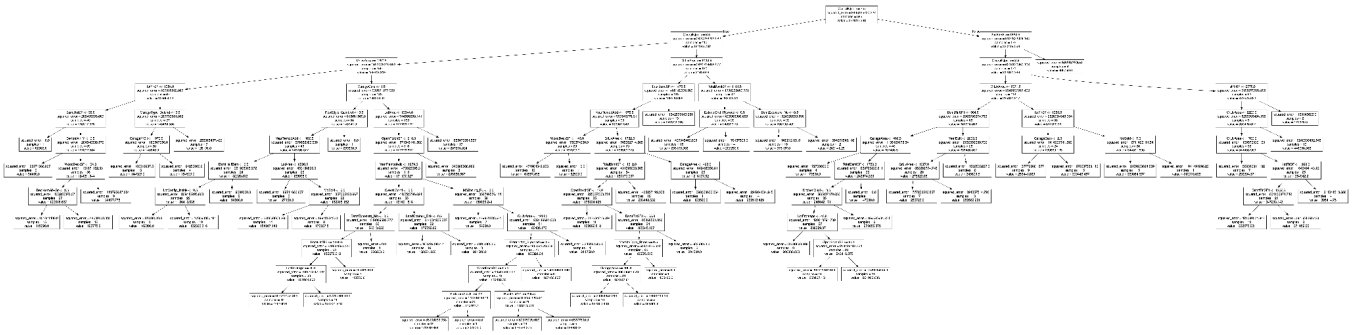


Ridge regression Alpha Vs. R2 (different range)

In steps of 1s, from 0 to 50, according to the computer,
Best alpha is 24
Highest R2 with this alpha is 0.6968942124531752

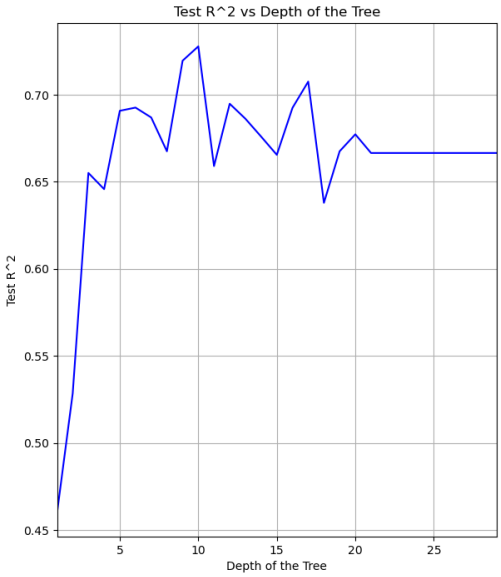
4. Regression trees

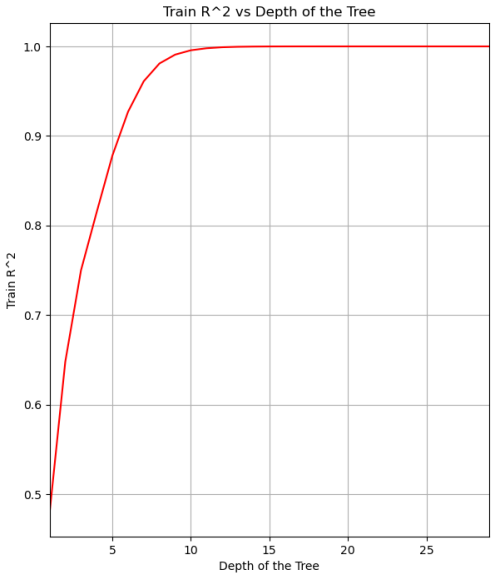
Unpruned tree visualization (see here for close view):



R^2 score of the unpruned model is: **0.6521199732905261.**

We have used 5-fold cross-validation search with **R2 score** to prune the tree at its optimal depth.





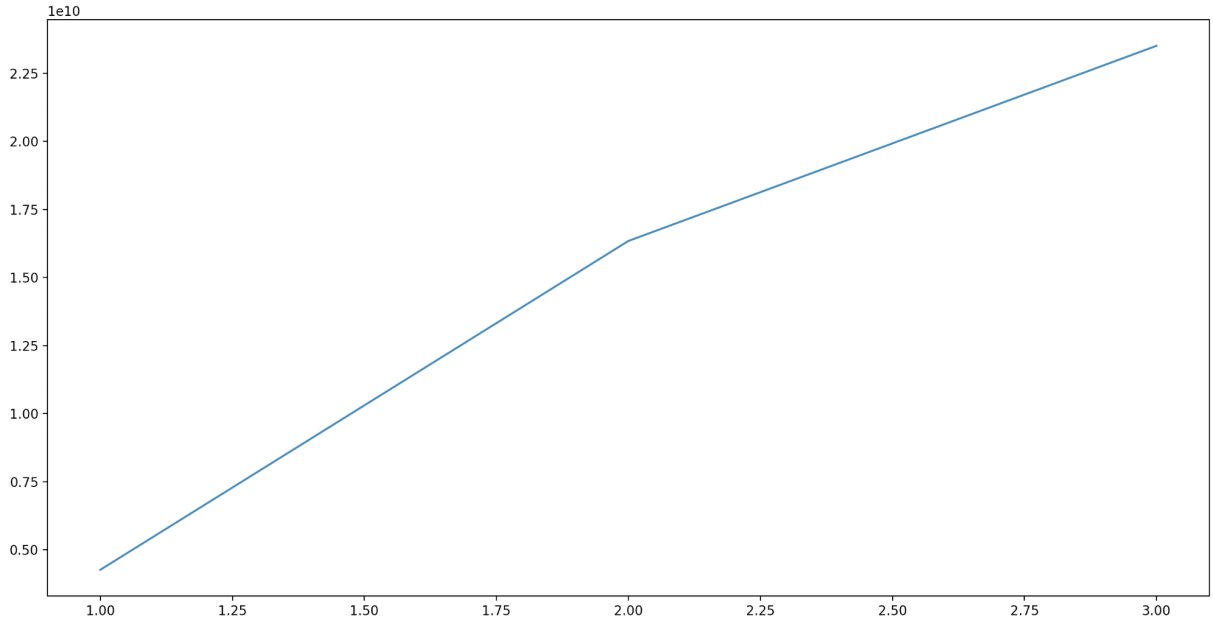
Best model in terms of CV MSE has **max depth 10**, with the increased 5-fold CV R^2 score of: **0.7277722320720575**.

Finally, we will use the Gradient Boosting and find the optimal shrinkage with cross-validation 5-folds. We find that with **learning rate 0.5** and **number of estimators of 29**, we get the optimal model with R^2 score of: **0.8435644149282833**.

5. Polynomial regression

Finding best degree:

To find the best degree, we take a look at the average test MSE through cross validation of each degree in list [1,2,3].



Average Test MSE vs. Polynomial degree

Polynomial degree 1: train MSE=590731544.6, test MSE=4263362071.49
Polynomial degree 2: train MSE =0.0, test MSE =16344695503.42
Polynomial degree 3: train MSE =0.0, test MSE =23513806009.42

As readers can tell, when the polynomial is going up, the training error is getting much smaller since we have a huge amount of predictors. And we also found that the polynomial fit is not the optimal since the average test MSE keeps going up.

Such a large MSE is expected, since there are lots of data points and the price for houses is usually pretty high.

Regression conclusion:

By comparing R^2 of each regression method, Ridge and Lasso have significant advantages over pure linear regression. And we would not want to consider a polyfeature regression since it only increases average test MSE while degree increases. The regression trees also achieve high accuracy with R^2 scores of 65% for unpruned model, 73% for the pruned and 84% for boosted one.

Therefore, if we want to predict the price of a house based on all information given in the dataset, we would choose boosted tree regression as our first choice, having the highest 84% R^2. Ridge or Lasso regression will be our second choice, since it has an overall good 70% R^2.

B - Predicting whether Sale Price will be above median

For this step we encoded the response variable as: 1 if Sale Price is above median and 0 otherwise.
Next we used forward stepwise feature selection to select the best number of predictors in the range between 1 and 40 predictors (more than that was too computationally expensive).
For `mlxtend.feature_selection's SequentialFeatureSelector`, we used a custom scoring function which calculates the error rate of the model. The model with the lowest error rate was selected.
To select among the best subsets with different number of predictors, we used BIC

$$BIC = -2 * LL + \frac{log(N)}{N} * k$$ [1]

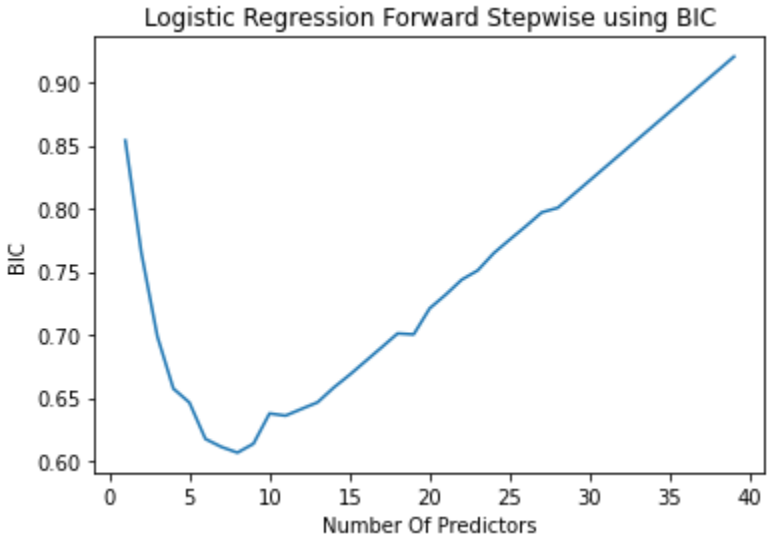
Where LL is the log-likelihood, N the number of observation and k the number of predictors.

Since log(N) >> LL for all models, we took the liberty of dividing the penalty term by N, which led to much improved results in all models.

Once the subset with the best number of predictors is selected, using 10-fold cross validation, a CV error rate is calculated.

6. Logistic Regression

Using forward stepwise selection and BIC scoring, we have obtained the following BIC vs number of predictors graph.

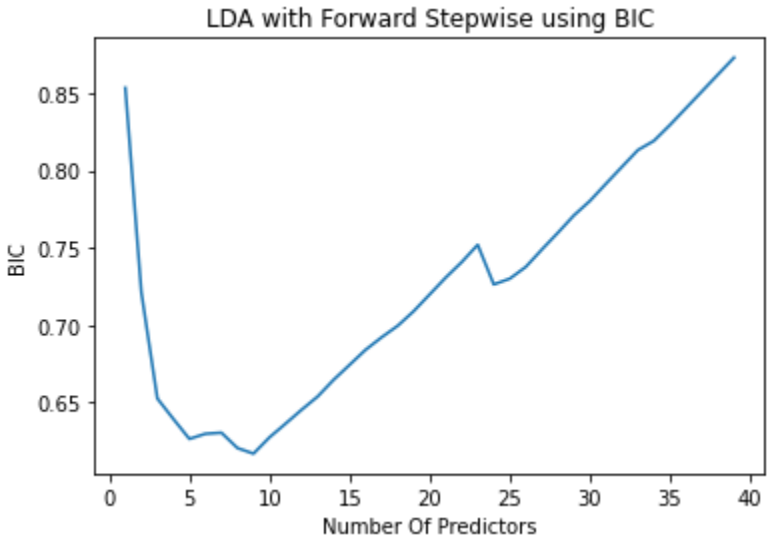


[1] https://machinelearningmastery.com/probabilistic-model-selection-measures/

From the graph we see that the best model using the BIC criterion was one with 8 predictors: `'GrLivArea', 'GarageCars', 'ExterQual', 'BsmtQual', 'MSZoning_FV', 'MSZoning_RL', 'LotShape_Reg', 'BsmtExposure_Gd'`

Using cross validation it was determined that the test error rate of a model using these 8 predictors and logistic regression was `0.09263588544710696`

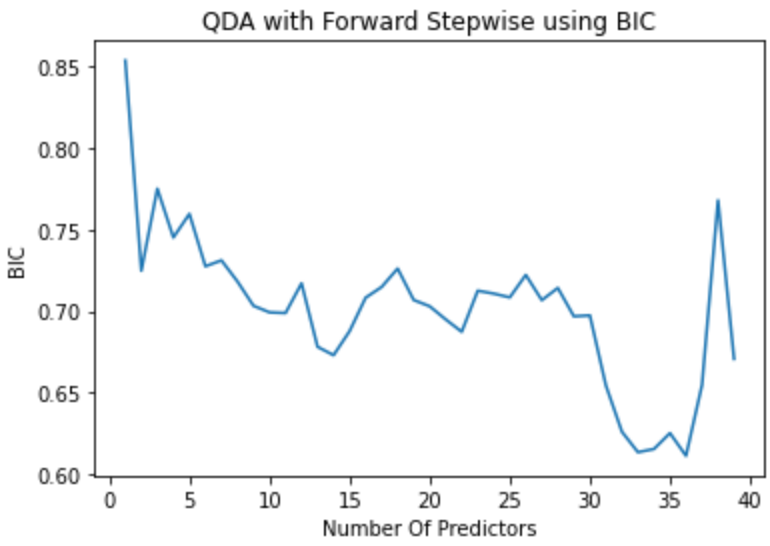
7. **Linear Discriminant Analysis (LDA) Classification**
   Using forward stepwise selection and BIC scoring, we have obtained the following BIC vs number of predictors graph.



From the graph we see that the best model using the BIC criterion was one with 9 predictors: `'OverallQual', 'OverallCond', 'GarageCars', 'ScreenPorch', 'ExterQual', 'MSZoning_RM', 'LandSlope_Mod', 'Exterior2nd_Brk Cmn', 'SaleCondition_Normal'`

Using cross validation it was determined that the test error rate of a model using these 8 predictors and logistic regression was `0.10265926358854471`

8. **Quadratic Discriminant Analysis (QDA) Classification**
   Using forward stepwise selection and BIC scoring, we have obtained the following BIC vs number of predictors graph.



From the graph we see that the best model using the BIC criterion was one with 36 predictors: `'LotFrontage', 'OverallQual', 'OverallCond', 'BsmtFinSF2', 'BsmtUnfSF', '1stFlrSF', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'GarageCars', 'WoodDeckSF', 'OpenPorchSF', 'ScreenPorch', 'MiscVal', 'MoSold', 'YrSold', 'BsmtQual', 'BsmtCond', 'MSSubClass_70', 'MSSubClass_75', 'MSSubClass_180', 'Alley_Pave', 'LandContour_Low', 'LandContour_Lvl', 'LandSlope_Gtl', 'BldgType_1Fam', 'Condition1_PosN', 'HouseStyle_SLvl', 'Exterior1st_BrkFace', 'Exterior2nd_HdBoard', 'MasVnrType_None', 'BsmtExposure_Mn', 'GarageFinish_RFn', 'SaleCondition_Abnorml'`

However, this increased flexibility and number of predictors did not help, as the cross-validation error rate was higher at `0.1385447106954997`

Classification conclusion:

We have found that with cross validation, the Logistic model is having the least error rate, less than 10%. On the other hand, with the LDA and QDA, we also get a relatively good result with an error rate slightly greater than 10%. We are satisfied with this result overall.