Home Sales Analysis (II) - Linear Models

In this analysis, we develop Linear Regression Models for the homes listed for sale in Colorado. The purpose is to do a deep dive into the nuts and bolts of the modeling process. Our approach is to:

> - develop insights into process of modeling for Linear Regression
> - develop a model that works well on unseen data

Why Linear Regression?
The key reason for focusing on Linear Regression is interpretability of such models. Inferences drawn from such models can be explained relatively easily.

Linear Model Selection

There are 168 odd predictor variables in this data set that can potentially influence the list price. Selecting a model that yields prediction accuracy as well as interpretability is of primary importance. [**ISLR, chapter 6, section 6.1.3**] discusses 3 classes of methods to approach this problem:

> - Subset Selection
> - Shrinkage
> - Dimension Reduction

Before, we delve into the methods, we point out that the above are relevant only in dealing with adding/removing variables in linear context. Hence are somewhat constrained. Other approaches for model building will include:
>
- examining higher order terms,
- examining interaction between variables,
- having a domain expert examine the variable set and help determine relevant variables for the model.

Subset Selection

This involves selecting a group of predictors from the available set of predictors based on pre-defined criteria. There are 3 primary methods of choosing the predictors:

> - Best subset selection
> - Stepwise selection
>   - Forward
>   - Backward
> - Hybrid approaches

Best subset selection

This is an exhaustive selection process where we fit regression models for each possible combination of predictor variables. Best model for each k (=1,2,…n) predictors is selected using largest R^2 or smallest RSS. Finally, among best model selected above, select the single best model using Cp, AIC, BIC or adjusted R^2.

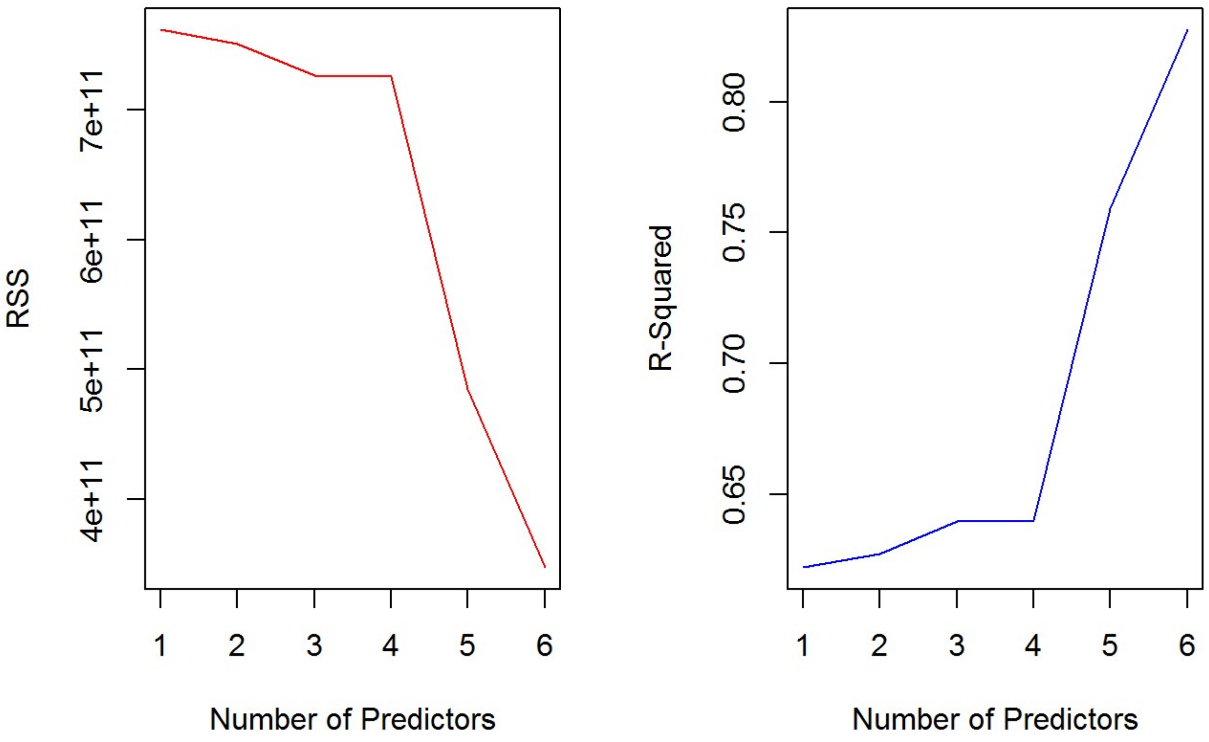
Of course, being an exhaustive selection process, it is mostly not feasible for all but smallest of the problems.

Note,as we increase the number of predictors in a model,

> - RSS decreases initially then flattens out
> - R^2 increases initially then flattens out

We demonstrate the above using our Housing Data:

```
##                                           Model R-Squared          RSS
## 1                                          SqFt 0.6221375 761695766543
## 2                                     SqFt+Beds 0.6274071 751073332372
## 3                               SqFt+Beds+Baths 0.6396677 726358298203
## 4                        SqFt+Beds+Baths+Spaces 0.6400694 725548650739
## 5            SqFt+Beds+Baths+Spaces+Street Type 0.7593894 485023266126
## 6 SqFt+Beds+Baths+Spaces+Street.Type+Style 0.8275315 347662322358
```

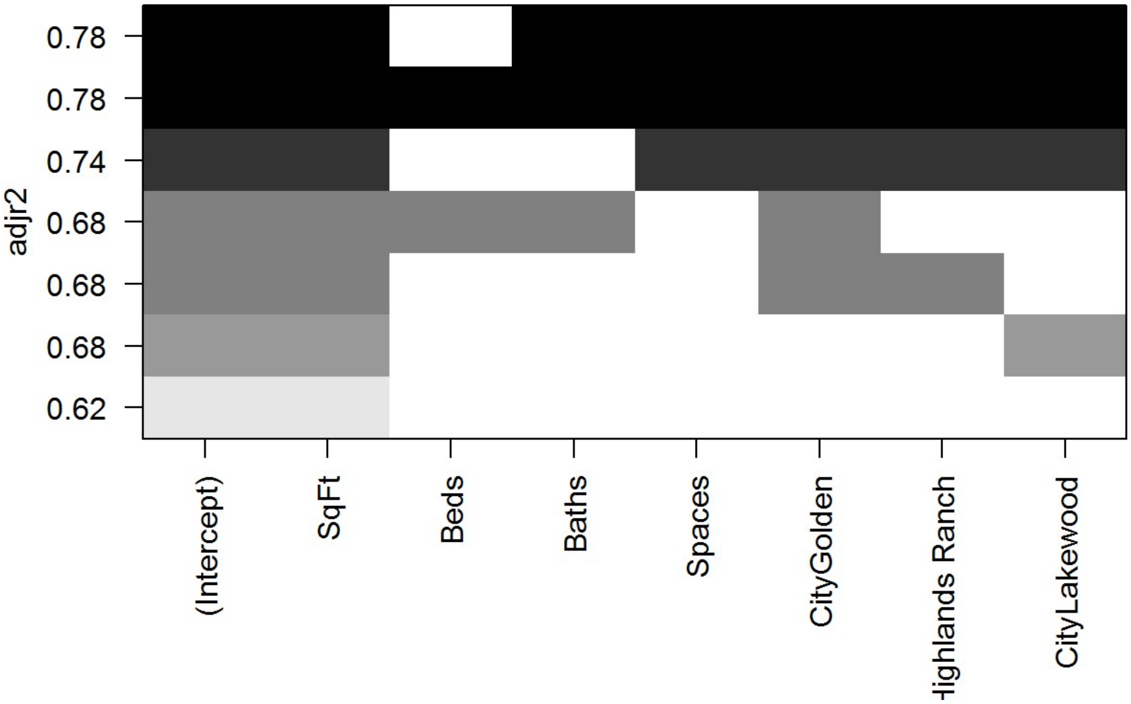


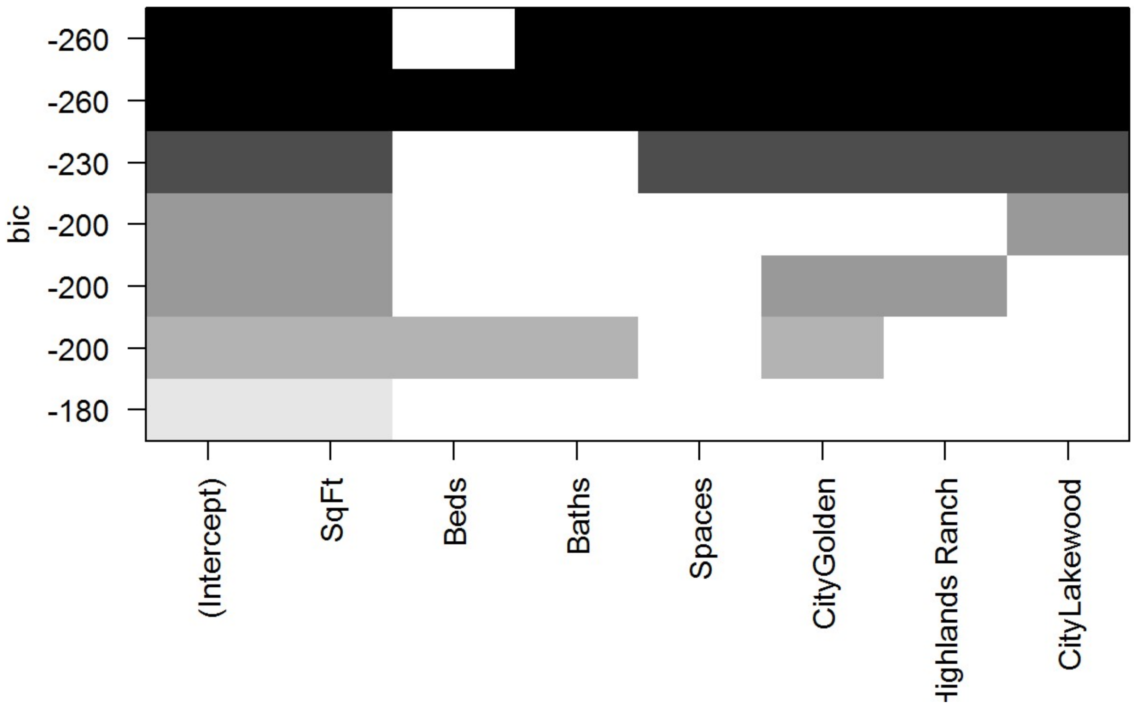
Note:

- Ideally, for *Best Subset Selection*, we should look at all possible models given the number of predictors. For example, 168 1-predictor models, (168c2) 2-predictor models, (168c3) 3-predictor models etc. Above, we work with ONE instance of n-predictor (n=1,2,3,4) models and make the comparisons between RSS and R^2 just to illustrate the conceptual point.
- Downsides: Computationally inefficient for large p. Prone to overfitting with high variance of coefficients.

All Subsets Selection in R

R provides function **regsubsets** for subset selection. In the plots below, we see that maximum adjusted R^2 and BIC is achieved by potentially leaving out **Beds** from the regression model.


0.78
0.78
0.74
0.68
0.68
0.68
0.62
adjr2
(Intercept)
SqFt
Beds
Baths
Spaces
CityGolden
ighlands Ranch
CityLakewood


bic
-260
-260
-230
-200
-200
-200
-180
(Intercept)
SqFt
Beds
Baths
Spaces
CityGolden
ighlands Ranch
CityLakewood

```
## Subset selection object
## Call: regsubsets.formula(Curr.List.Price ~ SqFt + Beds + Baths + Spaces +
##     City, data = data.2.na)
## 7 Variables  (and intercept)
##                      Forced in Forced out
## SqFt                    FALSE      FALSE
## Beds                    FALSE      FALSE
## Baths                   FALSE      FALSE
## Spaces                  FALSE      FALSE
## CityGolden              FALSE      FALSE
## CityHighlands Ranch     FALSE      FALSE
## CityLakewood            FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##          SqFt Beds Baths Spaces CityGolden CityHighlands Ranch
## 1  ( 1 ) "*"  " "  " "   " "    " "        " "
## 2  ( 1 ) "*"  " "  " "   " "    " "        " "
## 3  ( 1 ) "*"  " "  " "   " "    "*"        "*"
## 4  ( 1 ) "*"  "*"  "*"   " "    "*"        " "
## 5  ( 1 ) "*"  " "  " "   "*"    "*"        "*"
## 6  ( 1 ) "*"  " "  "*"   "*"    "*"        "*"
## 7  ( 1 ) "*"  "*"  "*"   "*"    "*"        "*"
##          CityLakewood
## 1  ( 1 ) " "
## 2  ( 1 ) "*"
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) "*"
## 6  ( 1 ) "*"
## 7  ( 1 ) "*"
```

Stepwise [Forward] Selection

Forward Selection is a hierarchical heuristic approach which drastically reduces the search space as compared to Subset Selection which is an exhaustive search process. For example, with 20 predictors, **Best Subset** requires fitting 1,048,576 models versus 211 models for **Forward selection** [**ISLR, pp 208**]. An intuitive explanation for developing a model with 4-predictors is as follows:

Model	Starting Model	Fixed Predictors	AvailAble Predictors	Possible Models	Selected
M0	-	-	-	Trivial with NO predictors	-

Model	Starting Model	Fixed Predictors	AvailAble Predictors	Possible Models	Selected
M1	M0	-	p1/p2/p3/p4	{M0+p1} OR {M0+p2} OR {M0+p3} OR {M0+p4}	p3
M2	M1	p3	p1/p2/p4	{M1+p1} OR {M1+p2} OR {M1+p4}	p2
M3	M2	p3 & p2	p1/p4	{M2+p1} OR {M2+p4}	p1
M4	M3	p3 & p2 & p1	p4	{M3+p4}	p4

> At each step,
> - select best model using min RSS or max R^2
> - final model selection between {M0, M1, M2, M3, M4} -using cross validated prediction error, Cp (AIC), BOC, or adjusted R^2.

Stepwise [Forward] Regression in R

```
## Start:  AIC=4147.52
## Curr.List.Price ~ SqFt + Beds + Baths + Spaces + City
##
##           Df  Sum of Sq        RSS    AIC
## - Beds     1 3.6552e+07 4.2524e+11 4145.5
## <none>                  4.2520e+11 4147.5
## - Baths    1 6.5934e+10 4.9114e+11 4173.2
## - Spaces   1 1.0475e+11 5.2995e+11 4187.8
## - SqFt     1 1.2862e+11 5.5382e+11 4196.3
## - City     3 3.0035e+11 7.2555e+11 4244.1
##
## Step:  AIC=4145.54
## Curr.List.Price ~ SqFt + Baths + Spaces + City
##
##           Df  Sum of Sq        RSS    AIC
## <none>                  4.2524e+11 4145.5
## + Beds     1 3.6552e+07 4.2520e+11 4147.5
## - Baths    1 7.8511e+10 5.0375e+11 4176.1
## - Spaces   1 1.0487e+11 5.3011e+11 4185.9
## - SqFt     1 1.2863e+11 5.5387e+11 4194.3
## - City     3 3.2016e+11 7.4540e+11 4247.3
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## Curr.List.Price ~ SqFt + Beds + Baths + Spaces + City
##
## Final Model:
## Curr.List.Price ~ SqFt + Baths + Spaces + City
##
##
##     Step Df Deviance Resid. Df   Resid. Dev      AIC
## 1                          184 425202645094 4147.521
## 2 - Beds  1 36552220       185 425239197314 4145.537
## Subset selection object
## Call: regsubsets.formula(Curr.List.Price ~ SqFt + Beds + Baths + Spaces + 
##     City, data = data.2.na, method = "forward")
## 7 Variables  (and intercept)
##                     Forced in Forced out
## SqFt                    FALSE      FALSE
## Beds                    FALSE      FALSE
## Baths                   FALSE      FALSE
## Spaces                  FALSE      FALSE
## CityGolden              FALSE      FALSE
## CityHighlands Ranch     FALSE      FALSE
## CityLakewood            FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: forward
##          SqFt Beds Baths Spaces CityGolden CityHighlands Ranch
## 1  ( 1 ) "*"  " "  " "   " "    " "        " "                
## 2  ( 1 ) "*"  " "  " "   " "    " "        " "                
## 3  ( 1 ) "*"  "*"  " "   " "    " "        " "                
## 4  ( 1 ) "*"  "*"  " "   " "    "*"        " "                
## 5  ( 1 ) "*"  "*"  " "   " "    "*"        "*"                
## 6  ( 1 ) "*"  "*"  " "   "*"    "*"        "*"                
## 7  ( 1 ) "*"  "*"  "*"   "*"    "*"        "*"                
##          CityLakewood
## 1  ( 1 ) " "         
## 2  ( 1 ) "*"         
## 3  ( 1 ) "*"         
## 4  ( 1 ) "*"         
## 5  ( 1 ) "*"         
## 6  ( 1 ) "*"         
## 7  ( 1 ) "*"
```

Stepwise [Backward] Regression

Similar to forward regression, backward regression is a heuristic. It starts with all predictors in the model and iteratively removes the least useful predictor.

If n < p, backward method cannot be used [**ISLR, pp. 204**]. Instead use Forward selection.

Choosing the Optimal Model

Typically R^2 and RSS are used to determine the goodness of a regression fit on a given data set. There are a couple of problems with this approach:

> - the model with all the predictors will always have the lowest RSS and highest R^2,
> - we are fitting models to given data set (train) and minimizing the error therein. What we really want is a model which minimizes the error on test data which the model has not seen before (test).
> - we note that, train error can be a poor estimate of test error.

There are couple of ways to approach this problem:

> - adjust the train error to assist in estimating the test error. This includes estimators such as: Cp, AIC, BIC, and adjusted R^2.
> - use validation set or cross validation approach.

We believe, former approach was widely used earlier when computation was expensive. This approach also had underlying assumptions which were needed to show that train error is an unbiased estimate of the test error. The validation/cross validation approaches which are now possible due to computational efficiencies help us look at the actual test error rather than an estimate based on the methods mentioned above. At the same time, they allow us circumvent the underlying assumptions.

Test Error estimators

> - **Cp** = (RSS + 2 * d *sigma(hat)^2)/n*
>   *where, sigma(hat)^2 - variance of epsilon and d - number of predictors*
>   *Term - 2* d * sigma(hat)^2 - can be considered to be a penalty proportional to d.
> - **AIC** = Cp / sigma(hat)^2
> - **BIC** = (RSS + log(n) * d * sigma(hat)^2), note: log(n) > 2 when n > 7, so heavier penalty for greater number of predictors
> - **Adjusted R^2** = 1 - [(RSS/(n-d-1)) / (TSS/(n-1))]
>   Now, maximizing R^2 => minimizing RSS/(n-d-1). Increasing d => increasing RSS/(n-d-1) => decreasing adjusted R^2
>   So, adjusted R^2 *pays a price* for increasing the number of predictor variables.

Cp, AIC, BIC have strong theoretical justifications.

Alternate - Validation/Cross Validation

These are resampling approaches which enable us to create train and test data sets thereby allowing a model developed using a train set to be validated against a test set. [*ISLR, chapter 5*]

References

[**ISLR**] - “An Introduction to Statistical Learning, with Applications in R”, James Gareth ET AL, Springer, Chapter 6, pp 203-259.