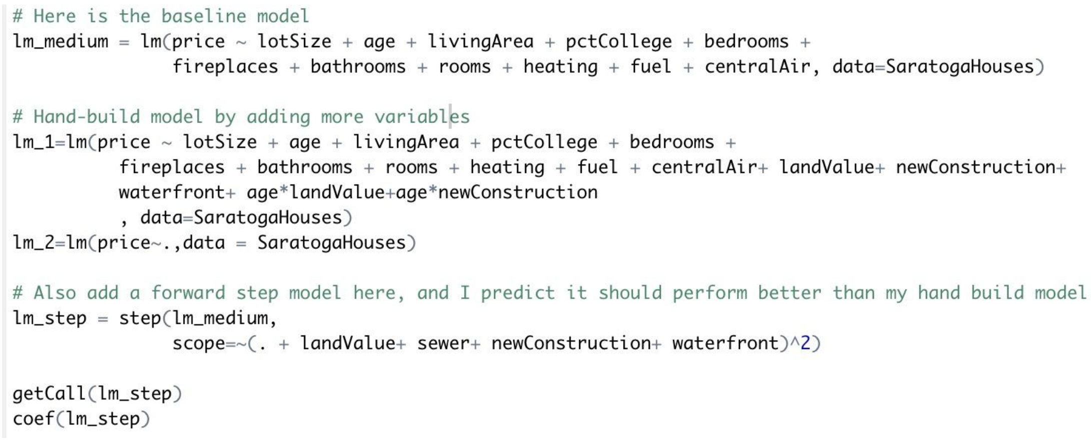
1. Saratoga house prices

The data set focused on the house price and different variables that might have influence on it. Here, as the question asked, I'm going to find a better model to predict the house price, based on the "medium" model which have been given. And also, to find some important variables that would contribute to this prediction. Then, turn this into a KNN model, and to make sure that I get a "right" KNN versus RMSE model by repeating some steps.

Since I am predicting the price-modeling strategies for a local taxing authority, who needs to form predicted market values for properties in order to know how much to tax them. I would like to find a most precise model to make sure the authority charging the right tax.

- Here are the models of predicting house price.



```
# Here is the baseline model
lm_medium = lm(price ~ lotSize + age + livingArea + pctCollege + bedrooms +
               fireplaces + bathrooms + rooms + heating + fuel + centralAir, data=SaratogaHouses)

# Hand-build model by adding more variables
lm_1=lm(price ~ lotSize + age + livingArea + pctCollege + bedrooms +
          fireplaces + bathrooms + rooms + heating + fuel + centralAir+ landValue+ newConstruction+
          waterfront+ age*landValue+age*newConstruction
          , data=SaratogaHouses)
lm_2=lm(price~.,data = SaratogaHouses)

# Also add a forward step model here, and I predict it should perform better than my hand build model
lm_step = step(lm_medium,
               scope=~(. + landValue+ sewer+ newConstruction+ waterfront)^2)

getCall(lm_step)
coef(lm_step)
```

Figure 1-1 Four models of predicting house price

To find how well these models predict the real data, I use RMSE to compare these models. And the less the number is, the better it fits the real data.

According to the result, except the original model 1, the forth model fits best with RMSE=59350.65, then the third (RMSE=61565.44), and the second model with interactions seems not work really well (RMSE=64278.01). But all of these model performs better then the "medium" model (RMSE=72755.4).

As a conclusion, the other variables in the data set also contributes in predicting the house price.

- And then, which variables should weight more than others to improve this prediction is important.

In the model, I squared each adding variables, and repeat the calculation of RMSE, and reach the final answers.

Compare with the original "medium" RMSE, both "sewer" and "new Construction" show nearly non-improve in RMSE, with the number 72775.07 and 72565.77. However, "land Value" which should influence the house price logically, not surprisingly, shows a great improvement in RMSE, which is 63865.26; "water front" also works to some extent (RMSE=69207.61).

- Now, try to fit this linear model to a KNN model to see if it could better predict the house price.

Using the right variables is really important in KNN models, so I delete the not helpful variables (sewer and new Construction) in the new model.

After the standardization, I use foreach loop to make sure I could get the optimal K value. And the KNN versus RMSE model prints below.

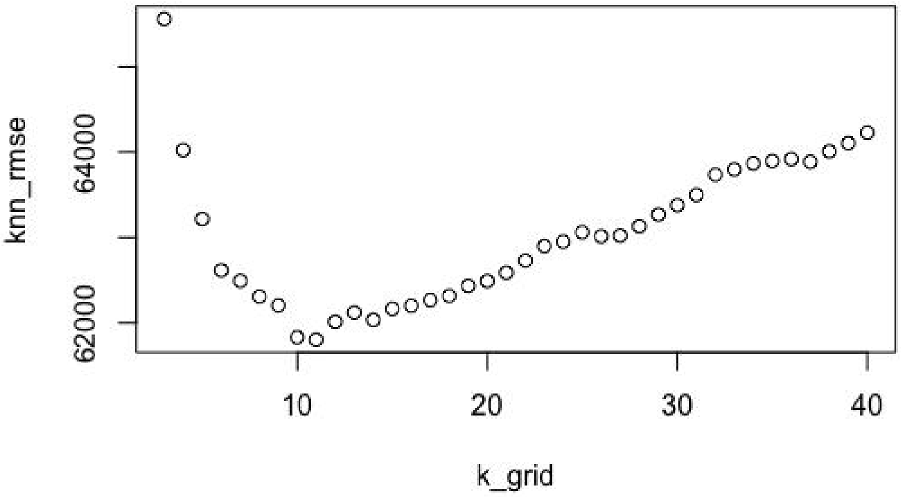
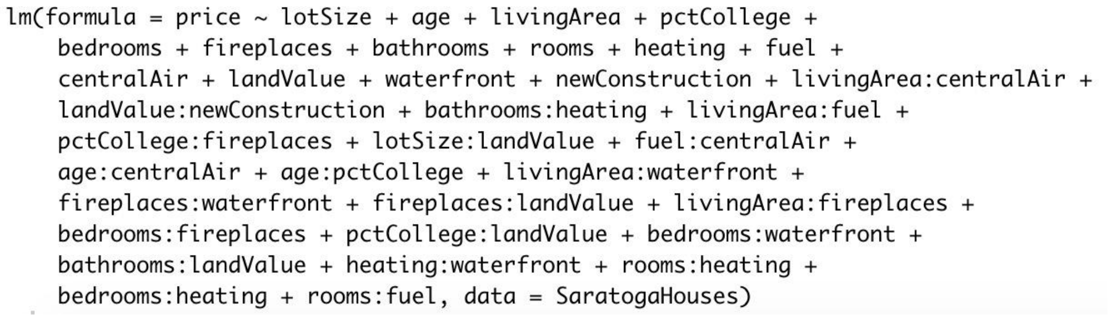


Figure 1-2 K vs RMSE plot

As the picture shows, the optimal K is 11, with a RMSE between 61500 and 62000, which seems not great compare with the hand-build linear regressions. Which means, the final result should combine these models together, and should also take the square part into consideration.

In my personal opinion, i would like to regard the forward step output as the best fits model, which is:

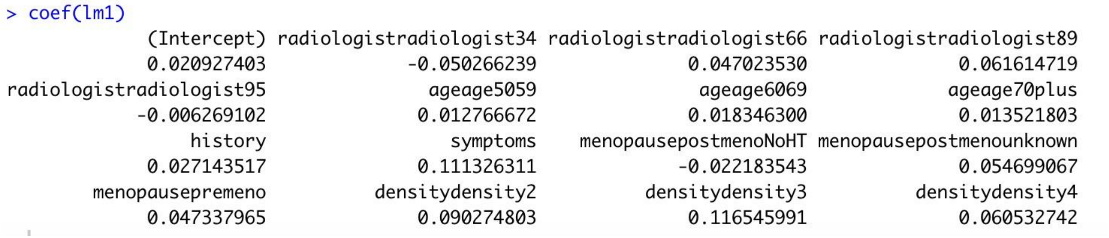


```
lm(formula = price ~ lotSize + age + livingArea + pctCollege +
    bedrooms + fireplaces + bathrooms + rooms + heating + fuel +
    centralAir + landValue + waterfront + newConstruction + livingArea:centralAir +
    landValue:newConstruction + bathrooms:heating + livingArea:fuel +
    pctCollege:fireplaces + lotSize:landValue + fuel:centralAir +
    age:centralAir + age:pctCollege + livingArea:waterfront +
    fireplaces:waterfront + fireplaces:landValue + livingArea:fireplaces +
    bedrooms:fireplaces + pctCollege:landValue + bedrooms:waterfront +
    bathrooms:landValue + heating:waterfront + rooms:heating +
    bedrooms:heating + rooms:fuel, data = SaratogaHouses)
```

2. A hospital audit

The goal for this session is to find if some radiologists are more conservative in recalling patients than others and also try to find important variables to improve the predict of cancer compare with what radiologists depend on now.

- First, I set a simple generalized linear model, which does not contain “cancer”, since “cancer” is a fixed true out put, and if I know some one who has cancer, I will definitely recall and provide further treatment. Thus, “cancer” could not be used as a factor in valuing “recall”.



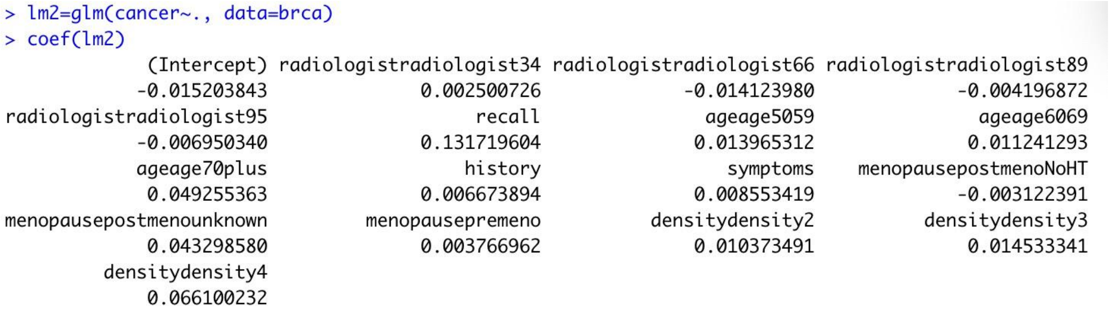
```
> coef(lm1)
            (Intercept) radiologistradiologist34 radiologistradiologist66 radiologistradiologist89
            0.020927403             -0.050266239              0.047023530              0.061614719
radiologistradiologist95                ageage5059               ageage6069              ageage70plus
           -0.006269102              0.012766672              0.018346300              0.013521803
                history                 symptoms    menopausepostmenoNoHT menopausepostmenounknown
            0.027143517              0.111326311             -0.022183543              0.054699067
        menopausepremeno           densitydensity2          densitydensity3          densitydensity4
            0.047337965              0.090274803              0.116545991              0.060532742
```

Figure 2-1 Coefficient in predicting “recall”

According to this linear regression, it’s easy to compare different radiologists recalling frequency holding other risk factors equal to a constant number-- under the same risk.

As it shows, ‘Intercept’ here represents “radiologists13”, and then I could get the recalling frequency of these four different radiologists, and the order is radiologist89 > radiologist66 > radiologist13 > radiologist34, from the most conservative to the least conservative person.

- Now, I would focus on which risk factors are important in predicting “cancer”, and also fit it in the recalling model.



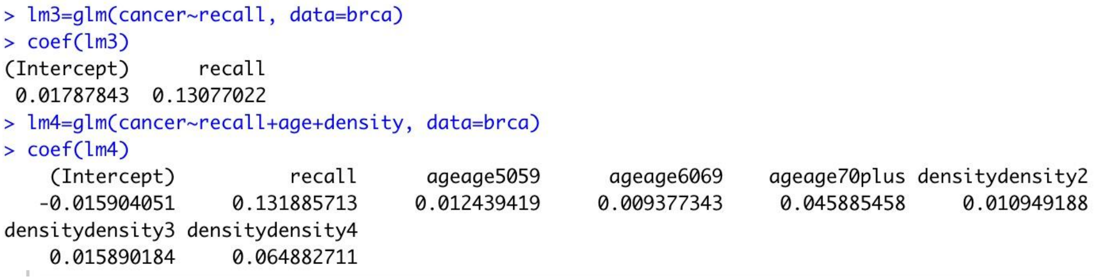
```
> lm2=glm(cancer~., data=brca)
> coef(lm2)
            (Intercept) radiologistradiologist34 radiologistradiologist66 radiologistradiologist89
           -0.015203843              0.002500726             -0.014123980             -0.004196872
radiologistradiologist95                   recall               ageage5059               ageage6069
           -0.006950340              0.131719604              0.013965312              0.011241293
           ageage70plus                  history                 symptoms    menopausepostmenoNoHT
            0.049255363              0.006673894              0.008553419             -0.003122391
menopausepostmenounknown         menopausepremeno          densitydensity2          densitydensity3
            0.043298580              0.003766962              0.010373491              0.014533341
         densitydensity4
            0.066100232
```

Figure 2-2 Coefficient in predicting “cancer”

Again, build a new model which regards “cancer” as the output, and see how much the factors could contribute.

It is obvious that except “recall”, “density4” and “age70plus” are significant with a really high slop, holding all else fixed. Thus, not only pay attention to or to simply improve the weight of “age” and “density”, it is more useful to focus on these two important subsets.

So while combining these factors and “recall” to predict “cancer”, we will get a higher coefficient between “recall” and “cancer” than only using “recall” to predict, and we just see if it works simply.

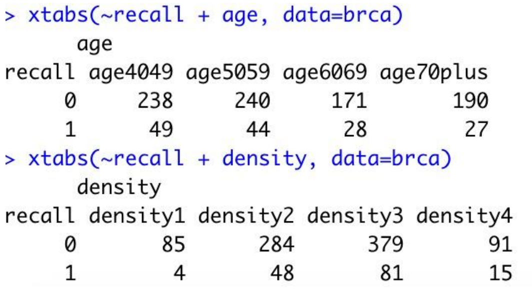


```
> lm3=glm(cancer~recall, data=brca)
> coef(lm3)
(Intercept)      recall 
 0.01787843  0.13077022 
> lm4=glm(cancer~recall+age+density, data=brca)
> coef(lm4)
    (Intercept)          recall      ageage5059      ageage6069    ageage70plus densitydensity2 
   -0.015904051     0.131885713     0.012439419     0.009377343     0.045885458     0.010949188 
densitydensity3 densitydensity4 
    0.015890184     0.064882711 
```

Figure 2-3 A improve in “recall” predict “cancer”

Again, from figure 2-1, we could easily notice that both these two factors are not highly weighted in recalling patients.

We could also gain this result from the real data.



```
> xtabs(~recall + age, data=brca)
      age
recall age4049 age5059 age6069 age70plus
     0     238     240     171       190
     1      49      44      28        27
> xtabs(~recall + density, data=brca)
      density
recall density1 density2 density3 density4
     0       85      284      379       91
     1        4       48       81       15
```

Figure 2-4 real recall numbers

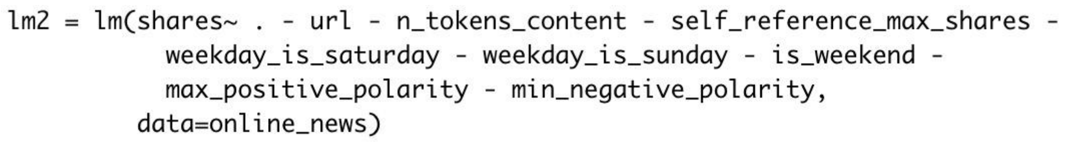
Both these two factors are not significantly noted by radiologists, no matter in the real number or in the proportion.

As a conclusion, doctors should add weight to the people age 70 plus, and those whose breast is extremely dense, while recalling people.

3. Predicting when articles go viral

Here I am trying to use the data to predict the important variables for a online news to be “viral”, which means to be shared more times. Also, to find the different between regress first and threshold second, and threshold first and regress/classify second, then we could get a certain way to deal with these data.

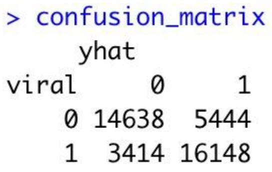
First, I set a simple linear model, and compare the p-value, then delete some not significant factors. The final linear regression shows below.



```
lm2 = lm(shares~ . - url - n_tokens_content - self_reference_max_shares -
           weekday_is_saturday - weekday_is_sunday - is_weekend -
           max_positive_polarity - min_negative_polarity,
         data=online_news)
```

Figure 3-1 Linear regression of predicting “shares”

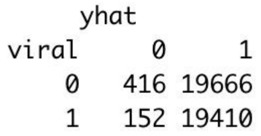
Then we could use this new linear model to make a prediction, and find how well it fits the real data, where share >1400. The performance of this model could be reported according to the confusion matrix, overall error rate, true positive rate, and false positive rate. To make sure, I use a loop here, and here is the output.



```
> confusion_matrix
     yhat
viral     0     1
    0 14638  5444
    1  3414 16148
```

Overall error rate = (5444+3414)/(14638+16148) ≈ 28.77% , true positive rate is 16148/(3414+16148) ≈ 82.55%, and false positive rate is 5444/(14638+5444) ≈ 27.11%.

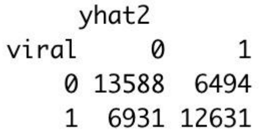
The baseline confusion matrix is



```
     yhat
viral     0     1
    0   416 19666
    1   152 19410
```

And the overall error rate is nearly 99.96%, a really bad model in prediction, and it means the new model really make sense in accurately predict the shares level.

Similarly, it is easy to get the fixed linear model in predicting viral. And this time I set it as a binary variable follow with the classification and regression.



```
     yhat2
viral     0     1
    0 13588  6494
    1  6931 12631
```

Overall error rate ≈ 51.26% , true positive rate ≈ 64.57%, and false positive rate ≈ 32.24%.

In my opinion, we should not do the threshold at first, since we will lose accuracy in predicting different level of the target variable, and also the numbers above show that it is not a good way to deal with data, which resulting in a much higher error rate and also lower TPR and higher FPR.