SDS Exercise 2

by Cheng Peng, Erich Schwartz, Zhiyuan Wei

1. KNN Practice

We evaluate the dataset on over 29,000 Mercedes S Class vehicles during 2014 and focus on three variables: price, mileage and trim level of cars for two specific models: 350 and 65 AMG. Our goal is to use K-nearest neighbors to build a predictive model for price, given mileage, separately for each of two trim levels: 350 and 65 AMG. We first create two datasets containing only the 350 model and the 65 AMG model. With train-test splits and applying different values of K, we report the root mean square error of each k and the relationship between RMSE and K as follows.

1.1 350 Model

RMSE for each value of K

$$k = 3 : RMSE_{out} = 1.1621 \times 10^4$$

$$k = 5 : RMSE_{out} = 1.0346 \times 10^4$$

$$k = 10 : RMSE_{out} = 9879$$

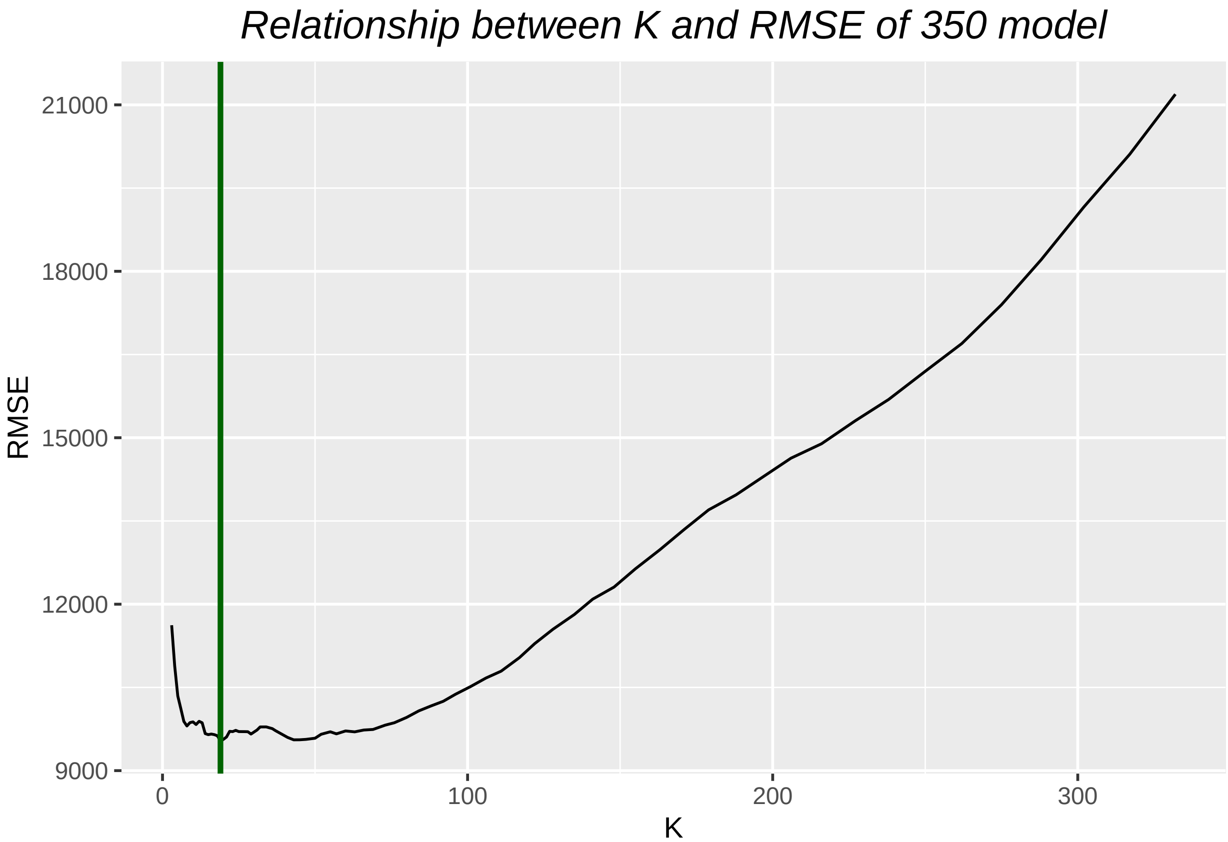
$$k = 20 : RMSE_{out} = 9566$$

$$k = 50 : RMSE_{out} = 9584$$

$$k = 100 : RMSE_{out} = 1.0505 \times 10^4$$

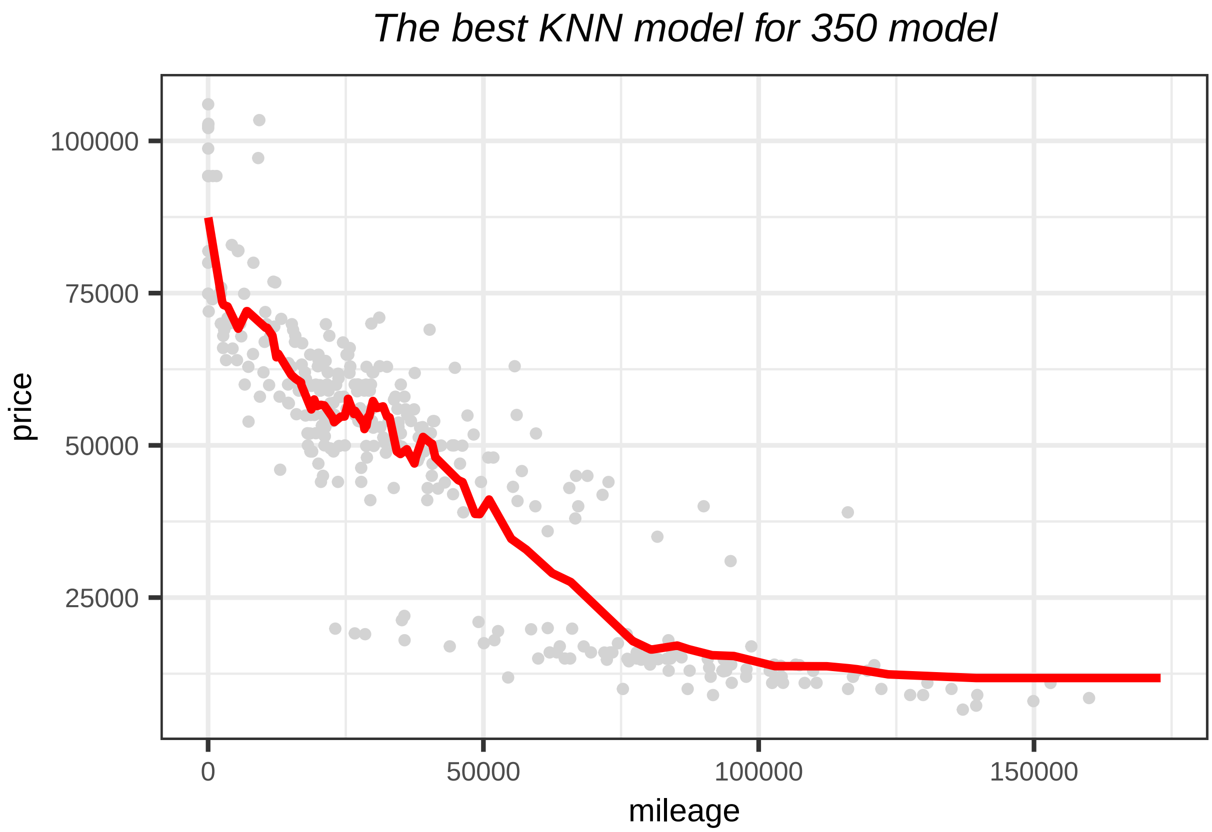
$$k = 332 : RMSE_{out} = 2.1192 \times 10^4$$

Plot the relationship between RMSE and K



$$The\ value\ of\ best\ k\ is:\ 19$$

Plot the best KNN model



$$RMSE_{out} = 9529$$

1.2 65 AMG Model

RMSE for each value of K

$$k = 3 : RMSE_{out} = 1.9339 \times 10^4$$

$$k = 5 : RMSE_{out} = 1.9289 \times 10^4$$

$$k = 10 : RMSE_{out} = 1.771 \times 10^4$$

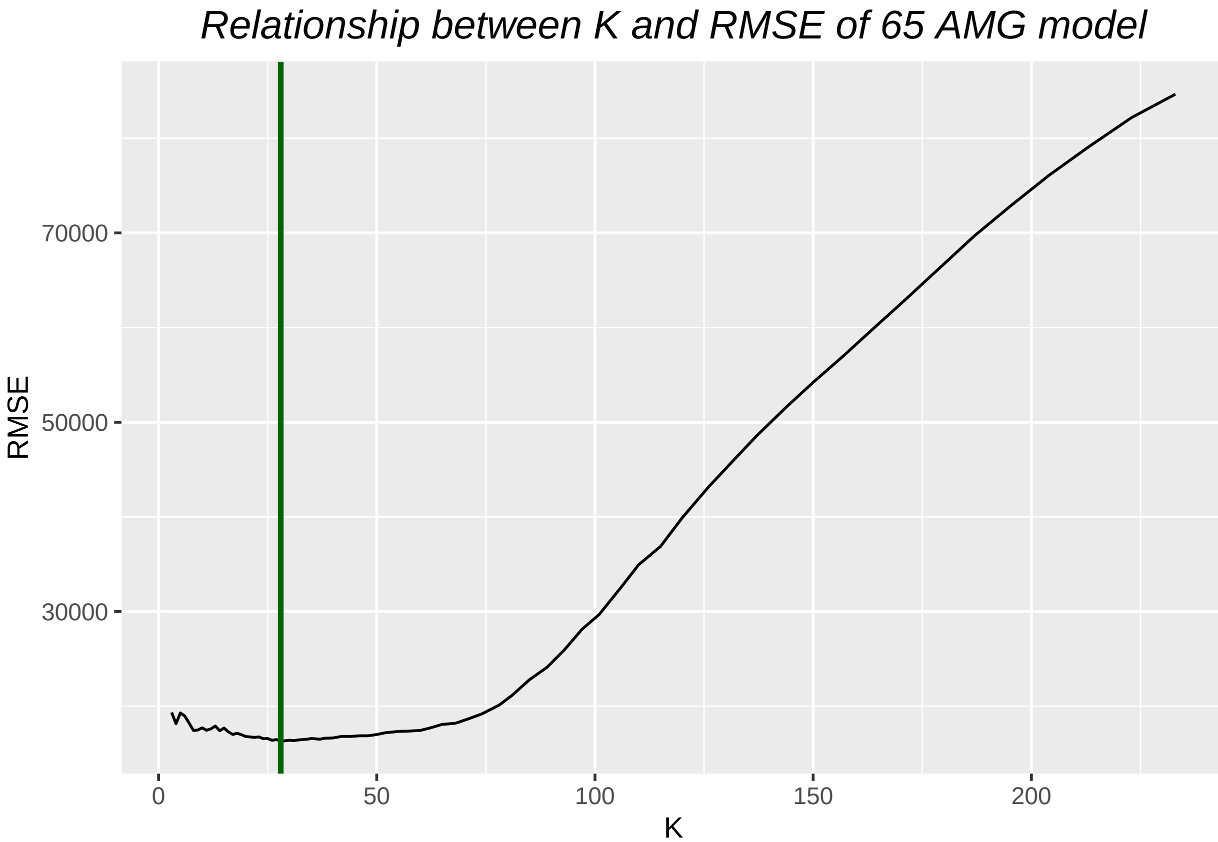
$$k = 20 : RMSE_{out} = 1.6792 \times 10^4$$

$$k = 50 : RMSE_{out} = 1.7006 \times 10^4$$

$$k = 100 : RMSE_{out} = 2.9254 \times 10^4$$

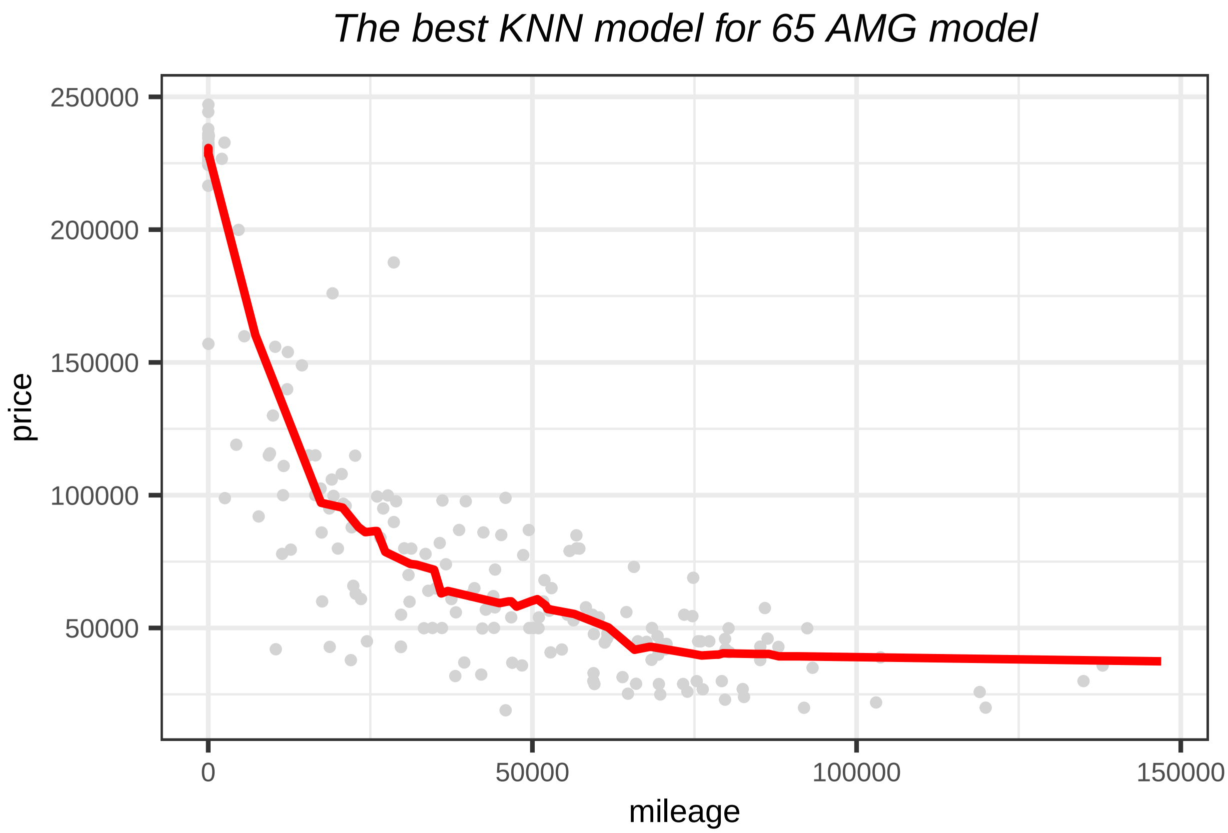
$$k = 233 : RMSE_{out} = 8.4659 \times 10^4$$

Plot the relationship between RMSE and K



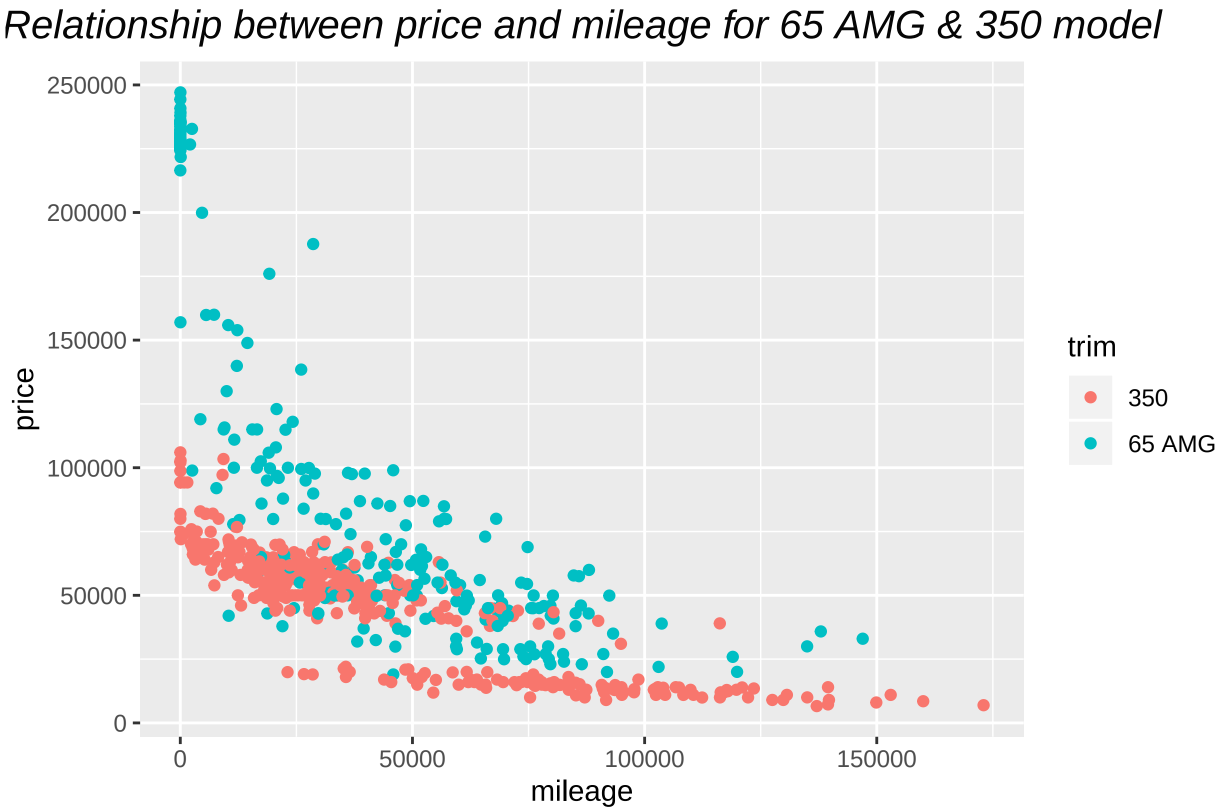
$$The\ value\ of\ best\ k\ is:\ 28$$

Plot the best KNN model



$$RMSE_{out} = 1.6294 \times 10^4$$

1.3 Conclusion



The price of 65 AMG models have greater variability than that of 350 model.

With repeated trials, we investigate the relationship between RMSE and K, plot the best KNN model for each of the two trim levels. We also find that the optimal value of K is generally higher for 65 AMG model than it is for 350 model. This seems unusual as the conventional wisdom states that K value is generally correlated with the sample size, and 350 model has more observations than the 65 AMG model.

We try to explain this phenomenon with the scatter plot above, which shows that the price of 65 AMG model has greater variance than that of 350 model. This implies the data of 65 AMG model has more "noise". And when larger values of k are used, the KNN algorithm reduces the noise in the data, which results in better out-of-sample performance. On the contrary, 350 model does not have much "noise". Therefore, a smaller k value returns better out-of-sample predictions.

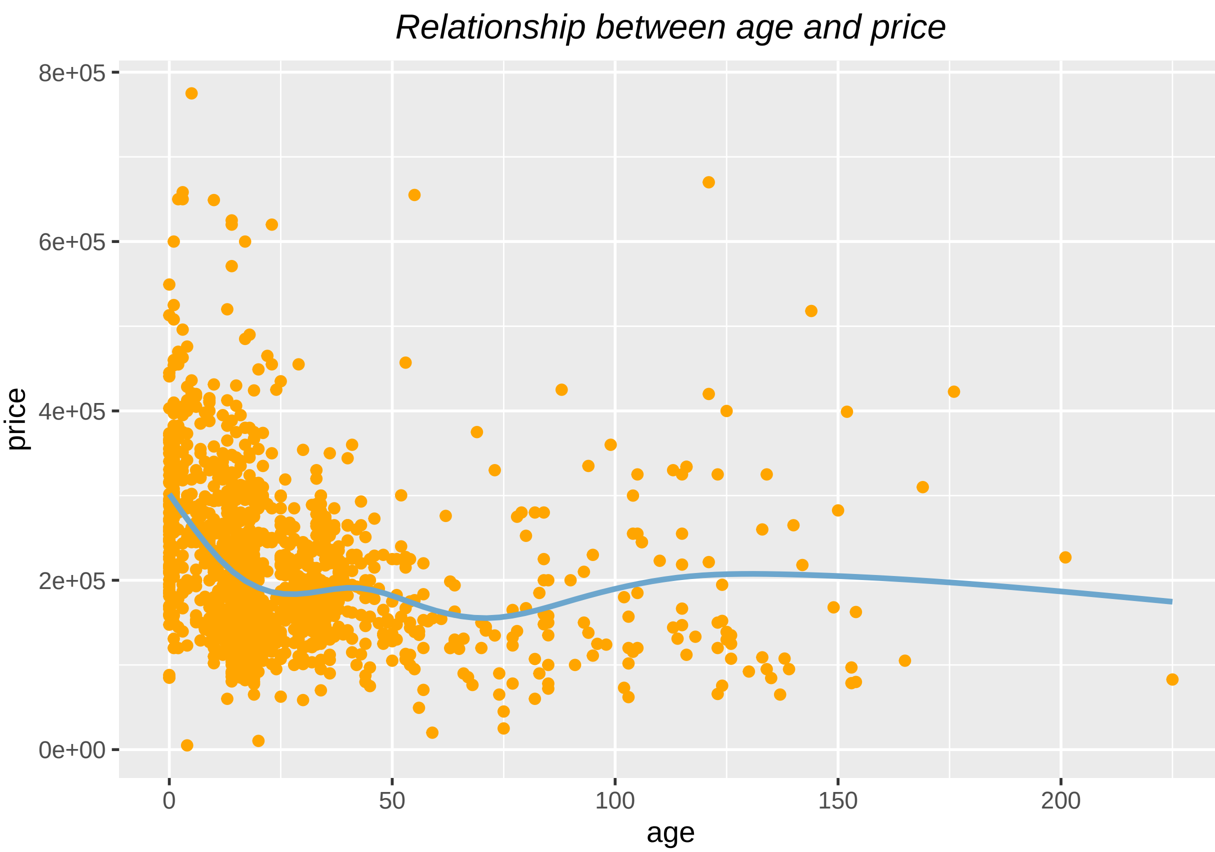
2. Saratoga house prices

From the dataset on house prices in Saratoga, NY, we evaluate the relations between price and different variables like number of rooms or size of parking lots. Our goal is to build a model that outperforms the "medium" model which includes 14 total variables, including the dummy variables to help taxing authorities determine how much to tax on properties by the predicted market values of properties. To investigate the relationship between house prices and key characteristics, we apply two methodologies: a hand-built linear model and a KNN model.

2.1 Hand-built model

In an attempt to create a better performing model, we first look into the relationship between house prices and several continuous variables to determine whether the relationship is linear or not. From real-life intuitions, we believe age to have a negative impact on the house price yet we expect a diminishing marginal impact of age on house prices as people are more sensitive to the ages of relatively new properties.

Transformation: Age and Price



The graph shows that age does seem to have a diminishing negative impact on price. Thus, to accommodate the diminishing marginal effect of age, we attempt two common types of transformation: log(age) and square root of age.

log(age) vs sqrt(age)

```
## [1] 63204.72 63325.12 63428.02
```

The root mean square error of using log transformation is 6.3205×10^4, while for square root transformation is 6.3325×10^4 and for no transformation is 6.3428×10^4. With the log transformation possessing the smallest RMSE, we adopt log transformation of age.

Finding variables & interactions with statistically significant impacts

We further investigate the impact of other discrete/continuous variables on house prices and incorporate statistically significant variables into our hand-built model.

```
##
## Call:
```

```
## lm(formula = price ~ . - sqrtage - age, data = saratoga_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -232992  -34712   -5460   28988  462733
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             1.372e+05  2.290e+04   5.994 2.62e-09 ***
## lotSize                 7.327e+03  2.820e+03   2.598 0.009475 **
## landValue               8.975e-01  5.409e-02  16.591  < 2e-16 ***
## livingArea              7.187e+01  5.022e+00  14.312  < 2e-16 ***
## pctCollege             -9.296e+01  1.674e+02  -0.555 0.578696
## bedrooms               -1.033e+04  2.841e+03  -3.636 0.000287 ***
## fireplaces             -5.163e+02  3.321e+03  -0.155 0.876491
## bathrooms               2.234e+04  3.726e+03   5.996 2.58e-09 ***
## rooms                   3.112e+03  1.049e+03   2.966 0.003069 **
## heatinghot water/steam -2.850e+03  4.620e+03  -0.617 0.537413
## heatingelectric        -3.794e+03  1.454e+04  -0.261 0.794242
## fuelelectric           -6.348e+03  1.441e+04  -0.441 0.659520
## fueloil                -3.816e+03  5.364e+03  -0.712 0.476884
## sewerpublic/commercial -3.103e+02  4.026e+03  -0.077 0.938582
## sewernone              -8.973e+03  1.692e+04  -0.530 0.595938
## waterfrontNo           -1.351e+05  1.857e+04  -7.277 5.72e-13 ***
## newConstructionNo       5.402e+04  8.987e+03   6.011 2.36e-09 ***
## centralAirNo           -1.060e+04  3.796e+03  -2.794 0.005287 **
## lgage                  -7.107e+03  1.941e+03  -3.661 0.000260 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56800 on 1363 degrees of freedom
## Multiple R-squared:  0.652,  Adjusted R-squared:  0.6474
## F-statistic: 141.8 on 18 and 1363 DF,  p-value: < 2.2e-16
```

From the regression results above, we find lotSize, landValue, livingArea, bathrooms, rooms, heatinghot water/steam, waterfront, newConstruction, CentralAir and lgage to be statistically significant at 5% significance level. Next, we go on to add the effects of interactions into our model by incorporating preceding variables and their interactions:

```
##
## Call:
## lm(formula = price ~ (lotSize + landValue + livingArea + bathrooms +
##     rooms + waterfrontNo + heatingHotwater + newConstruction +
##     centralAir + lgage)^2, data = saratoga_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -244746  -31830   -3814   28216  428178
##
## Coefficients: (2 not defined because of singularities)
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        7.687e+04  1.728e+05   0.445 0.656453
## lotSize                            2.619e+04  6.912e+04   0.379 0.704765
## landValue                          5.596e-01  4.617e-01   1.212 0.225740
```

```
## livingArea                          6.821e+01  1.004e+02   0.679 0.496949
## bathrooms                          -6.510e+03  8.935e+04  -0.073 0.941931
## rooms                               2.099e+04  1.827e+04   1.149 0.250879
## waterfrontNo                       -1.003e+05  1.627e+05  -0.617 0.537615
## heatingHotwater                     4.810e+04  6.579e+04   0.731 0.464868
## newConstructionNo                   7.113e+04  5.320e+04   1.337 0.181438
## centralAirNo                        1.775e+05  8.751e+04   2.028 0.042769 *
## lgage                              -3.042e+04  4.677e+04  -0.650 0.515501
## lotSize:landValue                  -3.625e-01  9.756e-02  -3.716 0.000211 ***
## lotSize:livingArea                 -1.790e+01  6.417e+00  -2.789 0.005365 **
## lotSize:bathrooms                   4.066e+03  4.861e+03   0.837 0.403021
## lotSize:rooms                       7.399e+02  1.685e+03   0.439 0.660706
## lotSize:waterfrontNo                2.042e+04  6.568e+04   0.311 0.755953
## lotSize:heatingHotwater             6.413e+03  6.342e+03   1.011 0.312132
## lotSize:newConstructionNo          -5.098e+03  1.912e+04  -0.267 0.789808
## lotSize:centralAirNo               -1.594e+04  7.458e+03  -2.137 0.032761 *
## lotSize:lgage                       2.446e+03  3.380e+03   0.724 0.469377
## landValue:livingArea               -1.466e-05  1.417e-04  -0.103 0.917611
## landValue:bathrooms                 1.137e-01  1.452e-01   0.783 0.433657
## landValue:rooms                    -5.042e-02  3.765e-02  -1.339 0.180745
## landValue:waterfrontNo              4.433e-02  2.764e-01   0.160 0.872607
## landValue:heatingHotwater          -6.991e-02  1.684e-01  -0.415 0.678049
## landValue:newConstructionNo         1.251e-01  2.358e-01   0.531 0.595846
## landValue:centralAirNo             -9.323e-03  1.436e-01  -0.065 0.948259
## landValue:lgage                     1.835e-01  6.390e-02   2.871 0.004153 **
## livingArea:bathrooms               -2.274e-01  5.986e+00  -0.038 0.969699
## livingArea:rooms                    2.915e+00  1.649e+00   1.767 0.077394 .
## livingArea:waterfrontNo             2.024e+01  9.172e+01   0.221 0.825418
## livingArea:heatingHotwater          9.965e+00  1.354e+01   0.736 0.461898
## livingArea:newConstructionNo        1.524e+01  3.602e+01   0.423 0.672296
## livingArea:centralAirNo            -2.036e+01  1.136e+01  -1.792 0.073352 .
## livingArea:lgage                   -1.309e+01  5.012e+00  -2.611 0.009127 **
## bathrooms:rooms                     6.563e+02  1.913e+03   0.343 0.731611
## bathrooms:waterfrontNo             -9.392e+03  8.402e+04  -0.112 0.911010
## bathrooms:heatingHotwater          -1.651e+04  1.053e+04  -1.567 0.117380
## bathrooms:newConstructionNo         1.275e+04  2.668e+04   0.478 0.632772
## bathrooms:centralAirNo             -4.161e+03  8.380e+03  -0.497 0.619590
## bathrooms:lgage                     7.078e+03  3.884e+03   1.823 0.068595 .
## rooms:waterfrontNo                 -1.913e+04  1.571e+04  -1.218 0.223348
## rooms:heatingHotwater              -2.341e+03  2.771e+03  -0.845 0.398271
## rooms:newConstructionNo            -7.434e+03  7.485e+03  -0.993 0.320802
## rooms:centralAirNo                 -1.700e+01  2.292e+03  -0.007 0.994082
## rooms:lgage                         7.185e+02  1.152e+03   0.624 0.533037
## waterfrontNo:heatingHotwater               NA         NA      NA       NA
## waterfrontNo:newConstructionNo             NA         NA      NA       NA
## waterfrontNo:centralAirNo          -1.287e+05  8.246e+04  -1.561 0.118789
## waterfrontNo:lgage                  4.469e+04  4.019e+04   1.112 0.266401
## heatingHotwater:newConstructionNo -2.148e+04  6.267e+04  -0.343 0.731839
## heatingHotwater:centralAirNo       -8.133e+03  1.409e+04  -0.577 0.563932
## heatingHotwater:lgage               1.821e+03  6.336e+03   0.287 0.773820
## newConstructionNo:centralAirNo     -3.251e+04  2.048e+04  -1.588 0.112568
## newConstructionNo:lgage            -2.971e+04  2.287e+04  -1.299 0.194240
## centralAirNo:lgage                  7.925e+03  4.265e+03   1.858 0.063341 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55160 on 1328 degrees of freedom
## Multiple R-squared:  0.6803, Adjusted R-squared:  0.6675
## F-statistic: 53.31 on 53 and 1328 DF,  p-value: < 2.2e-16
```

From the regression results above, we find landValue, heatingHotwater lotSize:landValue, lotSize:livingArea, lotSize:rooms, lotSize:centralAirNo, landValue:waterfrontNo, landValue:heatingHotwater, landValue:lgage, livingArea:bathrooms, livingArea:waterfrontNo, bathrooms:newConstructionNo, centralAirNo:lgage to be statistically significant at 10% confidence level. Therefore, we build a new model using the all the independent variables & preceding interactions.

```
##
## Call:
## lm(formula = price ~ landValue + lotSize + landValue + livingArea +
##     rooms + centralAir + waterfrontNo + heatingHotwater + lgage +
##     bathrooms + newConstruction + lotSize:landValue + lotSize:livingArea +
##     lotSize:rooms + lotSize:centralAir + landValue:waterfrontNo +
##     landValue:heatingHotwater + landValue:lgage + livingArea:bathrooms +
##     livingArea:waterfrontNo + bathrooms:newConstruction + centralAir:lgage,
##     data = saratoga_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -256141  -33094   -4730   30222  443345
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1.609e+05  6.734e+04   2.389 0.017036 *
## landValue                   5.814e-01  1.907e-01   3.049 0.002341 **
## lotSize                     5.011e+04  1.159e+04   4.325 1.64e-05 ***
## livingArea                  7.044e+01  3.256e+01   2.163 0.030688 *
## rooms                       1.224e+03  1.231e+03   0.995 0.320124
## centralAirNo               -3.339e+04  9.483e+03  -3.521 0.000444 ***
## waterfrontNo               -9.726e+04  4.944e+04  -1.967 0.049339 *
## heatingHotwater            -2.383e+03  6.320e+03  -0.377 0.706129
## lgage                      -1.990e+04  3.219e+03  -6.180 8.43e-10 ***
## bathrooms                  -7.227e+03  1.835e+04  -0.394 0.693717
## newConstructionNo           3.668e+04  4.410e+04   0.832 0.405641
## landValue:lotSize          -3.484e-01  8.066e-02  -4.319 1.68e-05 ***
## lotSize:livingArea         -1.255e+01  5.591e+00  -2.244 0.024976 *
## lotSize:rooms               5.341e+02  1.601e+03   0.334 0.738677
## lotSize:centralAirNo       -1.504e+04  5.892e+03  -2.553 0.010780 *
## landValue:waterfrontNo      6.021e-02  1.901e-01   0.317 0.751531
## landValue:heatingHotwater  -1.299e-01  1.506e-01  -0.862 0.388777
## landValue:lgage             1.619e-01  3.990e-02   4.059 5.22e-05 ***
## livingArea:bathrooms        1.362e+01  3.207e+00   4.246 2.32e-05 ***
## livingArea:waterfrontNo    -2.579e+01  3.169e+01  -0.814 0.415876
## bathrooms:newConstructionNo  5.090e+03  1.645e+04   0.309 0.757039
## centralAirNo:lgage          9.473e+03  3.123e+03   3.033 0.002467 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55740 on 1360 degrees of freedom
```

```
## Multiple R-squared:  0.6656, Adjusted R-squared:  0.6605
## F-statistic: 128.9 on 21 and 1360 DF,  p-value: < 2.2e-16
```

From the regression results above, we further eliminate interactions that are not statistically significant, namely, livingArea:waterfrontNo, bathrooms:newConstructionNo, landValue:waterfrontNo, lotSize:rooms.

```
##
## Call:
## lm(formula = price ~ lotSize + landValue + livingArea + rooms +
##     centralAir + waterfrontNo + heatingHotwater + lgage + bathrooms +
##     newConstruction + lotSize:landValue + lotSize:livingArea +
##     lotSize:centralAir + landValue:heatingHotwater + landValue:lgage +
##     livingArea:bathrooms + centralAir:lgage, data = saratoga_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -256471  -33193   -4690   30325  443371
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.830e+05  2.565e+04   7.136 1.56e-12 ***
## lotSize                    5.113e+04  1.090e+04   4.693 2.97e-06 ***
## landValue                  6.375e-01  1.077e-01   5.918 4.13e-09 ***
## livingArea                 4.491e+01  8.486e+00   5.292 1.41e-07 ***
## rooms                      1.478e+03  9.599e+02   1.539 0.123948
## centralAirNo              -3.400e+04  9.376e+03  -3.627 0.000298 ***
## waterfrontNo              -1.343e+05  1.800e+04  -7.458 1.55e-13 ***
## heatingHotwater           -2.393e+03  6.272e+03  -0.382 0.702812
## lgage                     -2.004e+04  3.176e+03  -6.312 3.72e-10 ***
## bathrooms                 -1.550e+03  6.681e+03  -0.232 0.816592
## newConstructionNo          5.007e+04  8.885e+03   5.636 2.11e-08 ***
## lotSize:landValue         -3.445e-01  7.786e-02  -4.424 1.05e-05 ***
## lotSize:livingArea        -1.121e+01  4.153e+00  -2.700 0.007016 **
## lotSize:centralAirNo      -1.476e+04  5.859e+03  -2.519 0.011882 *
## landValue:heatingHotwater -1.298e-01  1.491e-01  -0.870 0.384190
## landValue:lgage            1.626e-01  3.962e-02   4.104 4.30e-05 ***
## livingArea:bathrooms       1.326e+01  3.131e+00   4.234 2.45e-05 ***
## centralAirNo:lgage         9.641e+03  3.097e+03   3.113 0.001893 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55670 on 1364 degrees of freedom
## Multiple R-squared:  0.6654, Adjusted R-squared:  0.6613
## F-statistic: 159.6 on 17 and 1364 DF,  p-value: < 2.2e-16
```

Finally, from the regression results above, we eliminate variables heatingHotwater and bathrooms that are not statistically significant, and obtain our dominate model shown below.

Dominate model

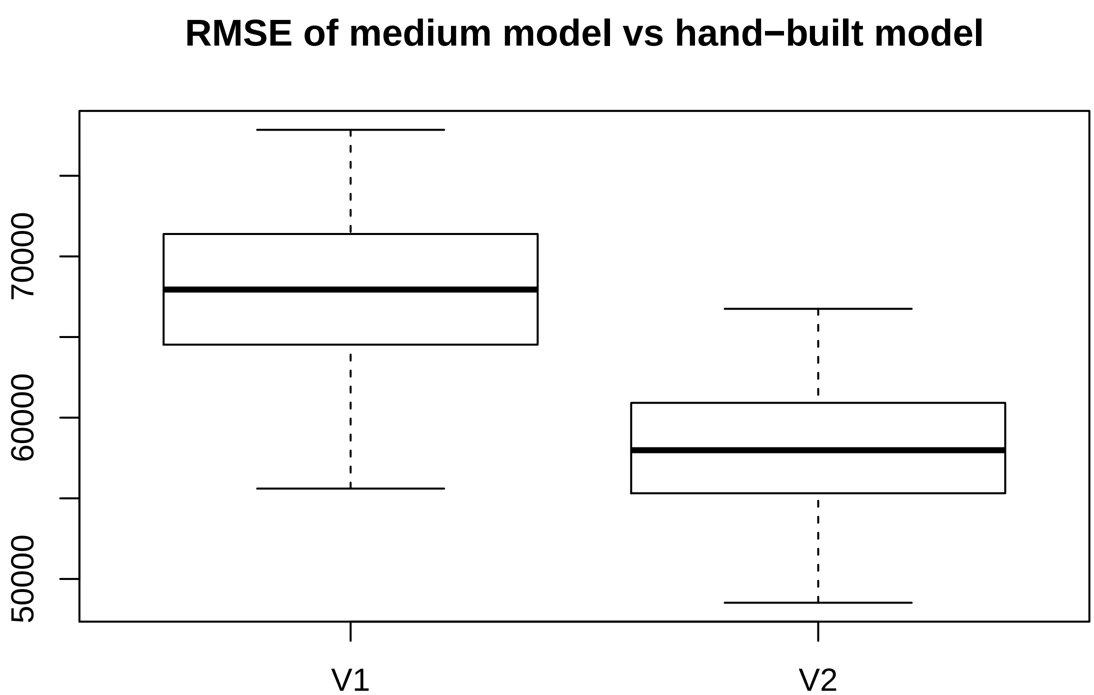
$$Price_{estimate} = \beta_0 \ + \ \beta_1 lotSize \ + \ \beta_2 landValue \ + \ \beta_3 livingArea \ + \beta_4 rooms \ + \beta_5 centralAir +$$
$$\beta_6 waterfrontNo \ + \ \beta_7 lgage \ + \ \beta_8 newConstuction \ + \ \beta_9 lotSize : landValue +$$

$$\beta_{10} lotSize : livingArea \ + \ \beta_{11} lotSize : centralAir \ + \beta_{12} landValue : heatingHotwater \ +$$
$$\beta_{13} landValue : lgage \ + \ \beta_{14} livingArea : bathrooms \ + \ \beta_{15} centralAir : lgage$$

```
##             (Intercept)               lotSize             landValue
##            1.812134e+05          5.084259e+04          6.343732e-01
##              livingArea                 rooms          centralAirNo
##            4.593445e+01          1.464295e+03         -3.424686e+04
##            waterfrontNo                 lgage     newConstructionNo
##           -1.342640e+05         -2.027415e+04          4.976885e+04
##       lotSize:landValue    lotSize:livingArea  lotSize:centralAirNo
##           -3.448770e-01         -1.109719e+01         -1.478173e+04
## landValue:heatingHotwater     landValue:lgage  livingArea:bathrooms
##           -1.720933e-01          1.676289e-01          1.259710e+01
##      centralAirNo:lgage
##            9.739152e+03
```

Comparison with the medium model

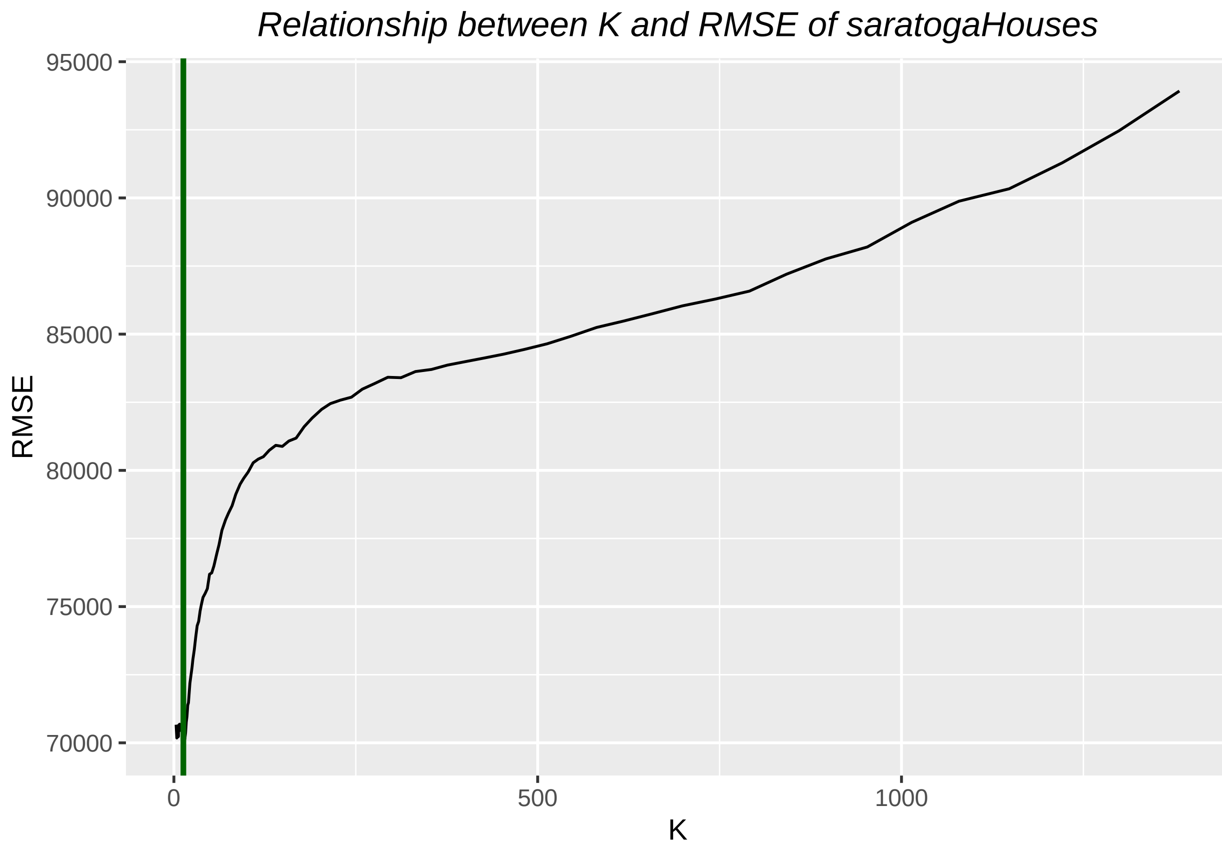
```
##       V1       V2
## 67750.93 58239.73
```

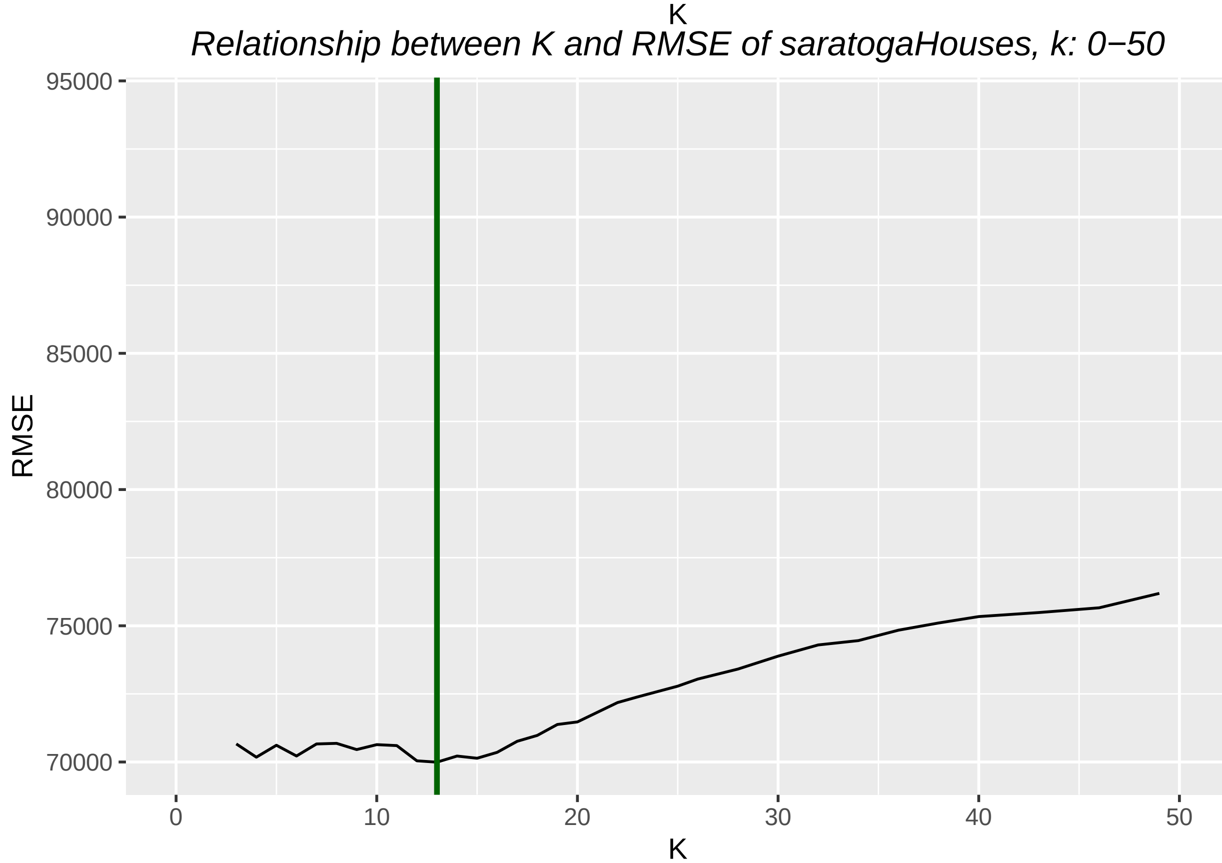


The boxplot above shows that our hand built model outperforms the medium model by 9511.19 in terms of the root mean square error.

2.2 KNN model

In order to construct the best KNN model, we engage in multiple train-test splits and try to find the optimum k value.





From the graphs above, we find that when k = 13, the KNN model have the lowest root mean square error.

The calculation shows that on average, the lowest root mean square error for KNN model is 6.6489×10^4, which is higher than the RMSEs of both the dominate model and the medium model.

2.3 Conclusions

We have evaluated the available data regarding homes in the Saratoga area with a view to devising a model to value those homes for tax purposes. In doing so, we have focused explicitly on statistical evidence of relationships between home features and home price that appear in the available data set. Certain of those relationships are not intuitive, but if the available data is sufficiently robust, the model taken as a whole should provide a reasonable prediction of home value.

The data that we considered included 15 characteristics of homes, which we evaluated to assess predictive value with regard to the price of the home. Certain of those characteristics did not appear to be meaningfully predictive, and were therefore not included in the model. We also considered interactions between factors, and identified several that had predictive value, including an interaction involving one of the features that was not independently predictive. Predictive value was assessed based on the statistical significance of the predictive relationship at a 95% confidence interval. We identified eight characteristics that were individually predictive, and seven characteristic interactions that were predictive.

100 randomly sampled training/test splits were performed utilizing that model and compared to a similar number of training/test splits utilizing the medium model against which we bench-marked our analysis. We found that the mean squared error of the predictions utilizing the new model was materially lower than the benchmark error, as reflected in the boxplot. The root mean square error in our new model is 5.9073×10^4 while in the medium model, the root mean square error is 5.3344×10^4. That analysis suggests that the new model will be more effective in predicting the value of homes for tax purposes.

We also build KNN models that predicts house prices with documented features. By plotting the relationship between the out-of-sample root mean square errors and k values, we find that the best k value is 4 and the corresponding root mean square error is 6.6489×10^4, which is higher than the root mean square error of our hand-built linear model. The best-performing KNN model also, surprisingly, has a higher out-of-sample root mean square error than the medium model.

1.There are six such characteristics: the percentage of college graduates, the number of bedrooms, fireplaces and bathrooms, type of heating fuel and the presence of a sewer connection. This may appear surprising since he number of bedrooms and bathrooms are ordinarily thought of as value predictors, but two features that are included in the model, the living area and number of rooms, are surrogates for those qualities that, based on our analysis, appear to be a more reliable predictor of value.
2.The eight individual features are lot size, land value, living area, number of rooms, central air conditioning, whether the property is waterfront, the age of the building (considered on a logarithmic scale) and whether it is new construction. Seven interactions, five of which were interactions with either lot size or land value, were also included in the model on that basis. Another feature, the type of heating, was not predictive standing alone, but was predictive in combination with the land value of a property and included in the model on that basis.

3. Predicting when articles go viral

We evaluated the data on 39,797 online articles published by Mashable between 2013 and 2014 and attempt to build models that successfully predict whether an article goes viral or not. More specifically, we focused on a few characteristics of online articles such as number of videos/images and stipulated that an article needed to be shared over 1400 times to be classified as 'viral'. In order to more accurately predict whether an article would go viral or not, we used two approaches: the classification approach and the regression approach. In classification approach, we first classified the training sets into two categories depending on whether the number of shares met the 1400 threshold. Then, we established a probability model for further predictions. In the regression approach, we estimated the number of shares for each distinct article, and based on the predicted shares, classified the training sets. Both methodologies are expected to outperform the null model yet we seek to find the best model under each approach and assess which approach yields a more accurate model in solving classification problems.

3.1 Classification Approach

We first built a new dataset from online_news but taking away the "url" variables and adding a new variable, "viral", which is a binary classification defining whether the article is viral or not depending on the variable "shares" (1 is viral with shares over 1400, 0 is not viral with shares less than or equal to 1400). The new dataset is called "online_news1".

After performing train-test splits on the new dataset, we built a baseline linear/logistic probability model with "viral" as dependent variable and all the remaining variables except "shares" as explanatory variables. Next, we applied the backward algorithm to identify those variables that are better at predicting whether an article goes viral in both models, and compared their in-sample performances.

Logistic Probability Model

```
##    yhat
## y       0     1
##   0 14289  1742
##   1 11118  4566
```

```
## [1] 0.5945136
```

Linear Probability Model

```
##    yhat
## y       0     1
##   0 10080  5951
##   1  5799  9885
```

```
## [1] 0.6295128
```

The in-sample performance of the best linear probability model is clearly better than the performance of the logistic probability model. The accuracy of the linear probability model is 0.63 while the accuracy of the logistic probability model is 0.595. Therefore, despite we might obtain false probabilities (>1 or <0) returned by linear probability models, for the sake of accurate classifications, we choose the linear probability model for further calculations.

Out-of-sample performance for linear probability model

```
##    yhat
## y      0    1
##   0 2519 1495
##   1 1450 2464
```

The above is the confusion matrix (where 0 represents non-viral articles while 1 represents viral ones; yhat stands for the model projections while y is the actual tabulations) obtained from the average of 100 out-of-sample performances of linear probability model. From which we can report that: The true positive rate is 0.629. The false positive rate is 0.373. The false discovery rate is 0.378. The overall accuracy rate is 0.628. The overall error rate is 0.372.

Comparison with the null model

```
##
##     0     1
## 16074 15641
```

From the table that tabulates viral/nonviral articles within the training set (where 0 represents the number of articles not viral and 1 represents the number of articles that are viral), it's reasonable to assume that "not viral" is the more likely outcome. So a reasonable null model is the one that guesses "not viral" for every test-set instance. Then, we investigate the out-of-sample performance for the null model.

```
##
##    0    1
## 4008 3921
```

```
## [1] 0.5054862
```

As demonstrated by the table above (where 0 represents the number of articles not viral and 1 represents the number of articles that are viral), in this particular train-test split, the accuracy of the null model is 0.505. With 100 train-test splits, our classification model returns an average accuracy rate of 0.628. Its absolute improvement over the null model is approximately 12.3 percent. Its relative improvement, or lift over the null model is 1.243. Clearly, our classification model demonstrates significant improvements of accuracy compared to the baseline model.

3.2 Regression Approach

log(shares) vs shares

We continue to use online_news1 as our dataset. However, instead of investigating variable "viral", when constructing baseline model, we use "shares"/"log(shares)" as our dependent variable and all the other variables except "viral" as explanatory variables. Next, we use backward algorithm to find the best linear model that predicts the shares/log transformation of shares with given features and compare the in-sample performance of the two models.

shares model

By conducting one train-test split, we obtain the in-sample performance of the model:

$$The\ accuracy\ of\ the\ linear\ model\ for\ log(shares)\ is\ 0.499$$

log(shares) model

By conducting one train-test split, we obtain the in-sample performance of the model:

$$The\ accuracy\ of\ the\ linear\ model\ for\ log(shares)\ is\ 0.586$$

Clearly, log(shares) model performs better at predicting whether an articles goes viral. Therefore, we employ log(shares) model in the following calculations and analysis.

Out-of-sample performance for the final regression model

```
##     yhat
## y       0    1
##   0 1307 2712
##   1  560 3350
```

The above is the confusion matrix (where 0 represents non-viral articles while 1 represents viral ones; yhat stands for the model projections while y is the actual tabulations) obtained from the average of 100 out-of-sample performances of regression model. From which we can report that: The true positive rate is 0.857. The false positive rate is 0.675. The false discovery rate is 0.447. The overall accuracy rate is 0.587. The overall error rate is 0.413.

Comparison with the null model

```
##
##    0    1
## 4009 3920
```

```
## [1] 0.5056123
```

We continue to assume the null model classifies all articles as "not viral". Then, In this particular train-test split. The accuracy of the null model is, as shown in the table above (where 0 represents the number of articles not viral and 1 represents the number of articles that are viral), 0.506. With 100 train-test splits, our regression model returns an average accuracy rate of 0.628. Its absolute improvement over the null model is approximately 12.287 percent. Its relative improvement, or lift over the null model is 1.243. The regression model shows a decent amount of improvements in accuracy over the null model.

3.3 Conclusions

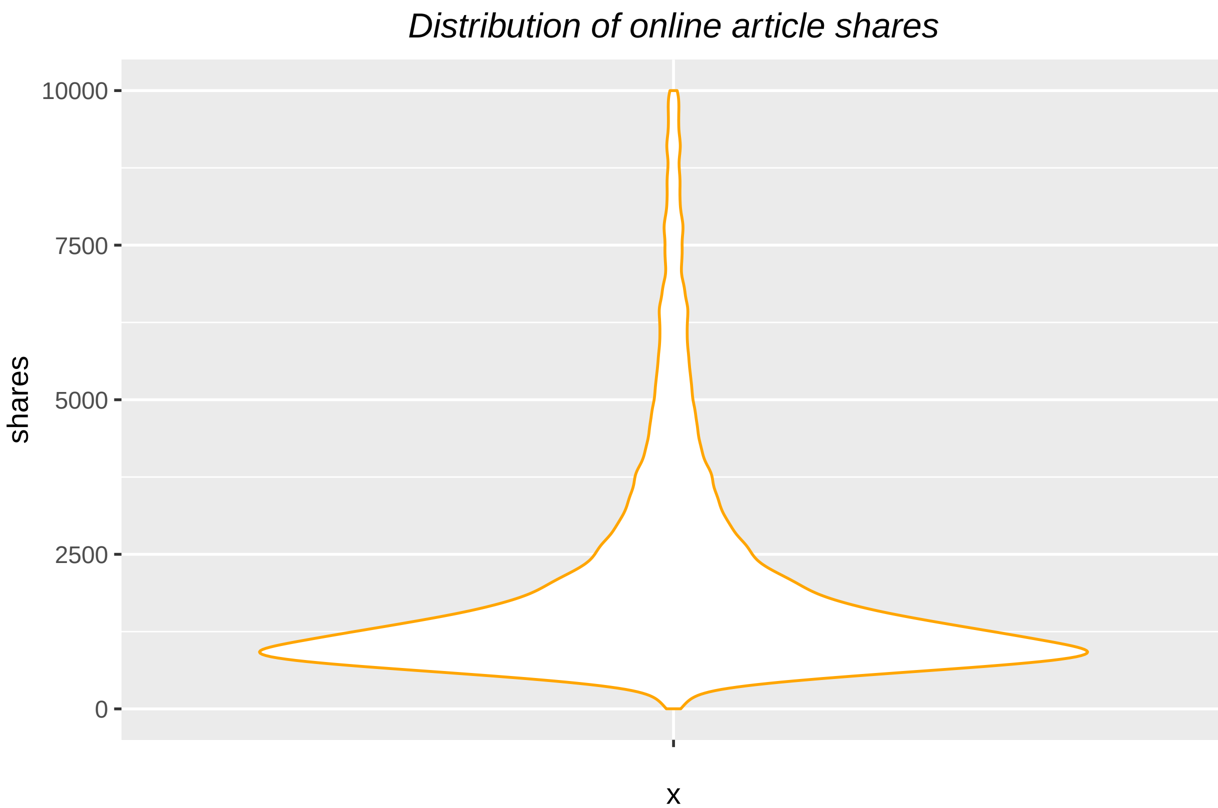
In order to identify characteristics that might predict whether an article goes "viral" using the definition provided, we utilized two distinct methodologies - classification and regression. We determined that a linear probability model performed better than the logistic probability model on our data set. We also discovered that using log transformation of shares in our regression model improves its out-of-sample accuracy.

Next, we compared the two best-performing models with the null model. 100 times of train/test splits were conducted. The average accuracy for the classification approach was 0.628 and 0.587 for the regression approach. While both of the final regression and classification models demonstrated decent amount of improvements over the null model accuracy of 0.506, classification models did seem to outperform the regression model by around 4%.

The statistical intuition behind the disparity, as we suspect, is connected to the fundamental difference between the two models. Classification directly predicts a binary condition (here, whether shares were below 1400 or not) whereas regression predicts a continuous quantity of shares. In classification, the model is agnostic as to whether an article is shared over 1 million times or is shared only 1401 times. In both cases, the article would be classified as "viral". As such, the classification method discards the information that isn't useful in mere classification. In regression models, however, the information of how many shares each article has is retained, which disrupts the goal of classifying. With large outliers in our dataset, the slope coefficients of explanatory variables are magnified. Consequently, the model tends to overestimate the exact number of shares. This leads to the high false positive rate as shown in previous discussions (0.675 for regression and 0.373 for classification) and undermines the accuracy of the regression model predictions.

Finally, from the model of the classification approach (technical details are shown below), we found that the data channel, day of publication, the length of the title, rate of positive/negative words in the content, minimum polarity of positive words, average polarity of positive/negative words, title subjectivity, and title polarity to have relatively greater impacts on whether an article goes viral. Based on the coefficients of the variables entailed, Mashable should: 1. Publish the article on Saturdays. 2. Have shorter titles. 3. Use more positive words and fewer negative words. 4. Use Social Media/Tech data channels 5. Use less polarized negative words but more polarized positive words. 6. Use more subjective and polarized title.

Supplementary graphs & classification model specifications



Many articles have online shares way above 1400, yet the exact numbers are not useful in classification.

```
##
## Call:
## lm(formula = viral ~ n_tokens_title + n_tokens_content + num_hrefs +
##     num_self_hrefs + num_imgs + average_token_length + num_keywords +
##     data_channel_is_lifestyle + data_channel_is_entertainment +
##     data_channel_is_bus + data_channel_is_socmed + data_channel_is_tech +
##     data_channel_is_world + self_reference_avg_sharess + weekday_is_monday +
##     weekday_is_tuesday + weekday_is_wednesday + weekday_is_thursday +
##     weekday_is_friday + weekday_is_saturday + avg_positive_polarity +
##     min_positive_polarity + avg_negative_polarity + min_negative_polarity +
##     title_subjectivity + title_sentiment_polarity, data = online_train)
##
## Coefficients:
##                  (Intercept)                 n_tokens_title
##                    7.139e-01                     -1.920e-03
##             n_tokens_content                      num_hrefs
##                    3.488e-05                      2.775e-03
##               num_self_hrefs                       num_imgs
```

```
##                 -5.927e-03                     1.624e-03
##       average_token_length                  num_keywords
##                 -2.855e-02                     9.718e-03
##    data_channel_is_lifestyle data_channel_is_entertainment
##                 -5.344e-02                    -2.109e-01
##          data_channel_is_bus        data_channel_is_socmed
##                 -7.221e-02                     1.395e-01
##         data_channel_is_tech         data_channel_is_world
##                  2.109e-02                    -2.212e-01
##   self_reference_avg_sharess             weekday_is_monday
##                  9.463e-07                    -1.485e-01
##           weekday_is_tuesday          weekday_is_wednesday
##                 -1.798e-01                    -1.772e-01
##          weekday_is_thursday             weekday_is_friday
##                 -1.674e-01                    -1.244e-01
##          weekday_is_saturday         avg_positive_polarity
##                  4.882e-02                     8.905e-02
##        min_positive_polarity         avg_negative_polarity
##                 -1.219e-01                    -9.626e-02
##        min_negative_polarity            title_subjectivity
##                  2.679e-02                     2.245e-02
##     title_sentiment_polarity
##                  4.603e-02
```