SDS 323: Exercises 2 Report

Samuel Higgins Rylan Keniston Rithvik Saravanan

March 13, 2020

KNN practice

Many car retail companies around the world strive to provide their customers with accurate and relevant market-based pricing information for vehicles. Certain types of car makes, such as Mercedes-Benz, prove to be especially difficult to predict pricing information. In the case of Mercedes-Benz, the Mercedes S class provides a challenging task for predictions because it is a very broad range of sub-models that all have the label “S Class”. This category includes vehicles ranging from luxury sedans to high-performance sports cars. Additionally, the individual submodels consist of cars with various different features. Due to this wide variety of factors, retail companies often struggle to provide accurate pricing predictions to consumers for these types of vehicles.

The data that we analyzed for this case includes more than 29,000 Mercedes S Class vehicles. To build our predictive model of price, we focused on three particular variables:

- *trim*: categorical variable for car’s trim level, e.g. 350, 63 AMG, etc. The trim is like a sub-model designation.
- *mileage*: mileage on the car
- *price*: the sales price in dollars ($) of the car

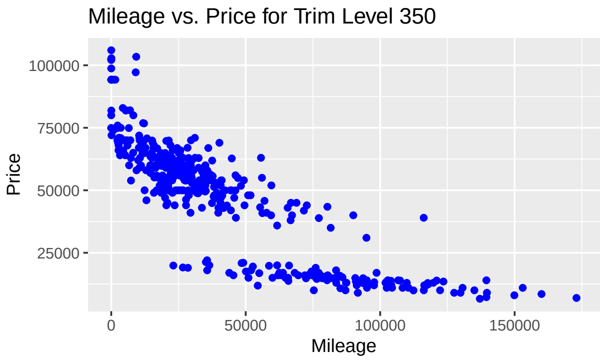
In addition to these variables, the data set includes several other useful values, such as:

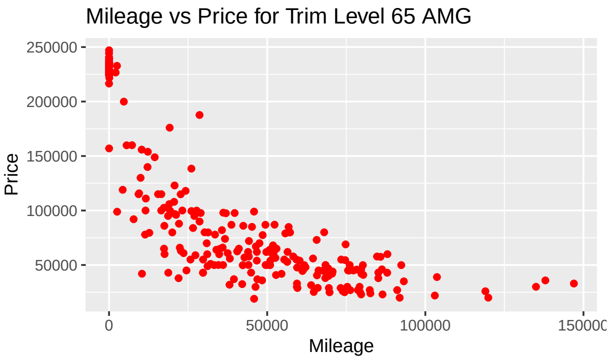
```
##   id trim subTrim condition isOneOwner mileage year  color displacement
## 1  2  320    unsp      Used          f  129948 1995   Gold        3.2 L
## 2  4  320    unsp      Used          f  140428 1997  White        3.2 L
## 3  7  420    unsp      Used          f  113622 1999 Silver        4.2 L
## 4  8  420    unsp      Used          f  167673 1999 Silver        4.2 L
## 5 11  500    unsp      Used          f   63457 1997 Silver        5.0 L
## 6 13  430    unsp      Used          f   82419 2002  White        4.3 L
##       fuel state region soundSystem wheelType wheelSize featureCount price
## 1 Gasoline    PA    Mid     Premium     Alloy      unsp           26  6595
## 2 Gasoline    NY    Mid        Bose     Alloy      unsp           22  7993
## 3 Gasoline    NJ    Mid        unsp     Alloy      unsp           24  5995
## 4 Gasoline    GA    SoA        unsp     Alloy      unsp           24  3000
## 5 Gasoline    CO    Mtn      Alpine     Alloy        20           23 14975
## 6 Gasoline    NJ    Mid        Bose     Alloy        16           35  7400
```

For more information on this data set, here is a summary of its data:

```
##        id              trim          subTrim       condition    isOneOwner
##  Min.   :    2   550    :21836   Hybrid:  190   CPO : 3586   f:25340
##  1st Qu.:13231   430    : 2071   unsp  :29276   New :10317   t: 4126
##  Median :26254   500    : 2002                  Used:15563
##  Mean   :26269   63 AMG : 1413
##  3rd Qu.:39293   600    :  527
##  Max.   :52572   350    :  416
##                  (Other): 1201
##     mileage            year          color        displacement
##  Min.   :     1   Min.   :1988   Black  :12838   4.6 L  :13599
##  1st Qu.:    14   1st Qu.:2007   Silver : 6095   5.5 L  : 9154
##  Median : 26120   Median :2012   White  : 4418   4.3 L  : 2071
##  Mean   : 40387   Mean   :2010   Gray   : 2007   5.0 L  : 2002
##  3rd Qu.: 68234   3rd Qu.:2015   Blue   : 1599   6.0 L  :  403
##  Max.   :488525   Max.   :2015   unsp   : 1467   6.3 L  :  391
##                                  (Other): 1042   (Other): 1846
##        fuel          state          region           soundSystem
##  Diesel  :  312   CA     : 5262   SoA    :7805   Alpine         :    2
##  Gasoline:28628   FL     : 3559   Pac    :5844   Bang Olufsen   :  177
##  Hybrid  :  189   NY     : 2754   Mid    :5824   Bose           :  943
##  unsp    :  337   TX     : 2458   WSC    :2865   Boston Acoustic:    1
##                   NJ     : 2266   ENC    :2496   Harman Kardon  : 4120
##                   GA     : 1408   New    :1421   Premium        : 9694
##                   (Other):11759   (Other):3211   unsp           :14529
##    wheelType       wheelSize      featureCount       price
##  Alloy  :14565   unsp   :25293   Min.   :  0.00   Min.   :   599
##  Chrome :   80   18     : 1774   1st Qu.: 18.00   1st Qu.: 28995
##  Premium:  424   19     : 1297   Median : 53.00   Median : 56991
##  Steel  :   49   20     :  813   Mean   : 46.48   Mean   : 67001
##  unsp   :14348   17     :  149   3rd Qu.: 70.00   3rd Qu.:108815
##                  16     :  107   Max.   :132.00   Max.   :299000
##                  (Other):   33
```

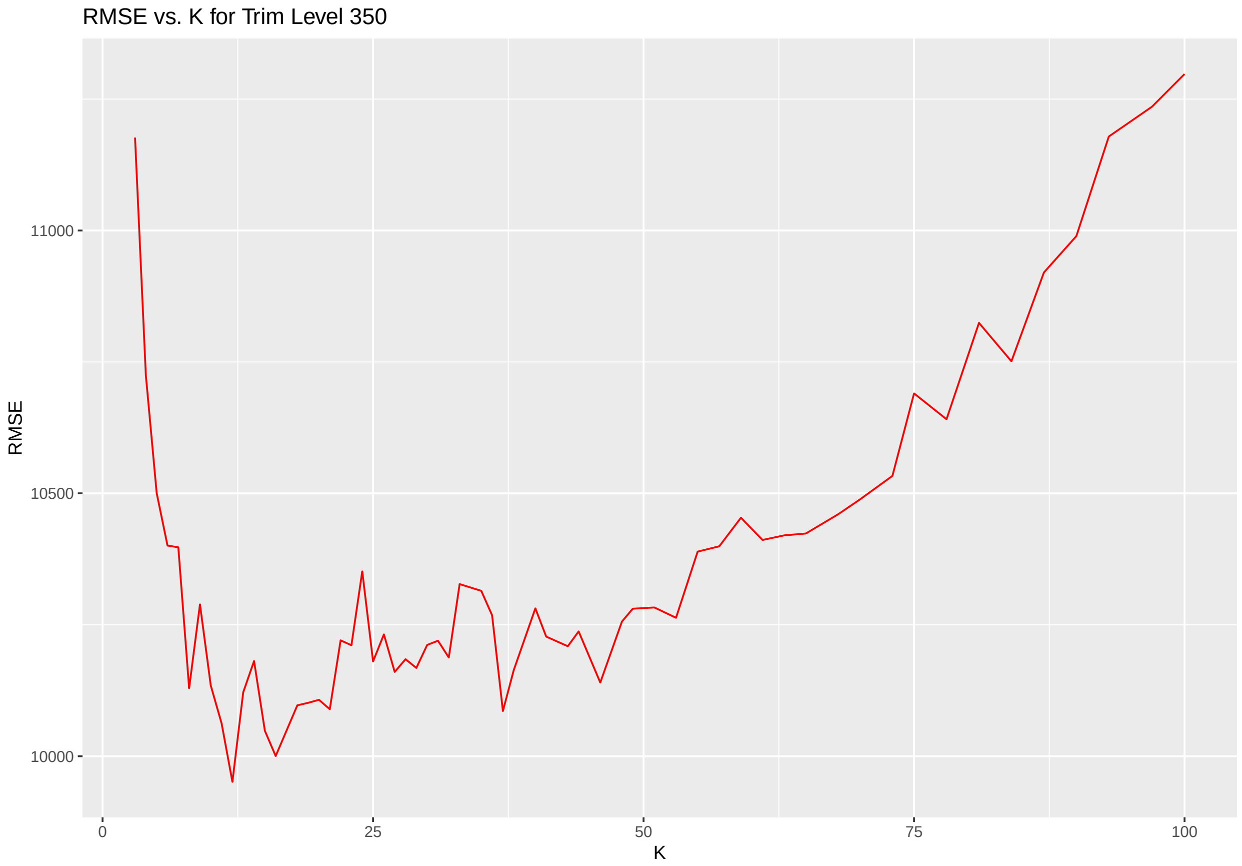
In this analysis, we are primarily focusing on two trim levels of the Mercedes S Class vehicles: 350 and 65 AMG. On the “Mileage vs. Price for Trim Level 350” plot below, it is interesting to note that these is a sizable gap in the data points in the price range $\$25,000$ to $\$35,000$. This indicates that there were likely very few types of cars in the Mercedes S class with the 350 trim that were sold for this price range, especially since almost all of the data points are either above or below this range. This type of gap is not present for the 65 AMG trim vehicles. This is an important factor because this may affect our KNN model since some data points may have “more distant” k nearest neighbors than other data points, which would skew the price predictions for those data points. Another interesting point from these plots is that the 350 trim data is more dense and concentrated in two distinct price ranges while the 65 AMG trim is less dense and more evenly distributed across the whole price range. An important insight as to why these distinctions occur is that the 350 trim is a middle-level luxury sedan while the 65 AMG trim is a upper-level sports car. This explains why the price range for the 350 trim is significantly lower than that of the 65 AMG trim (up to about $125,000 vs. up to about $250,000).



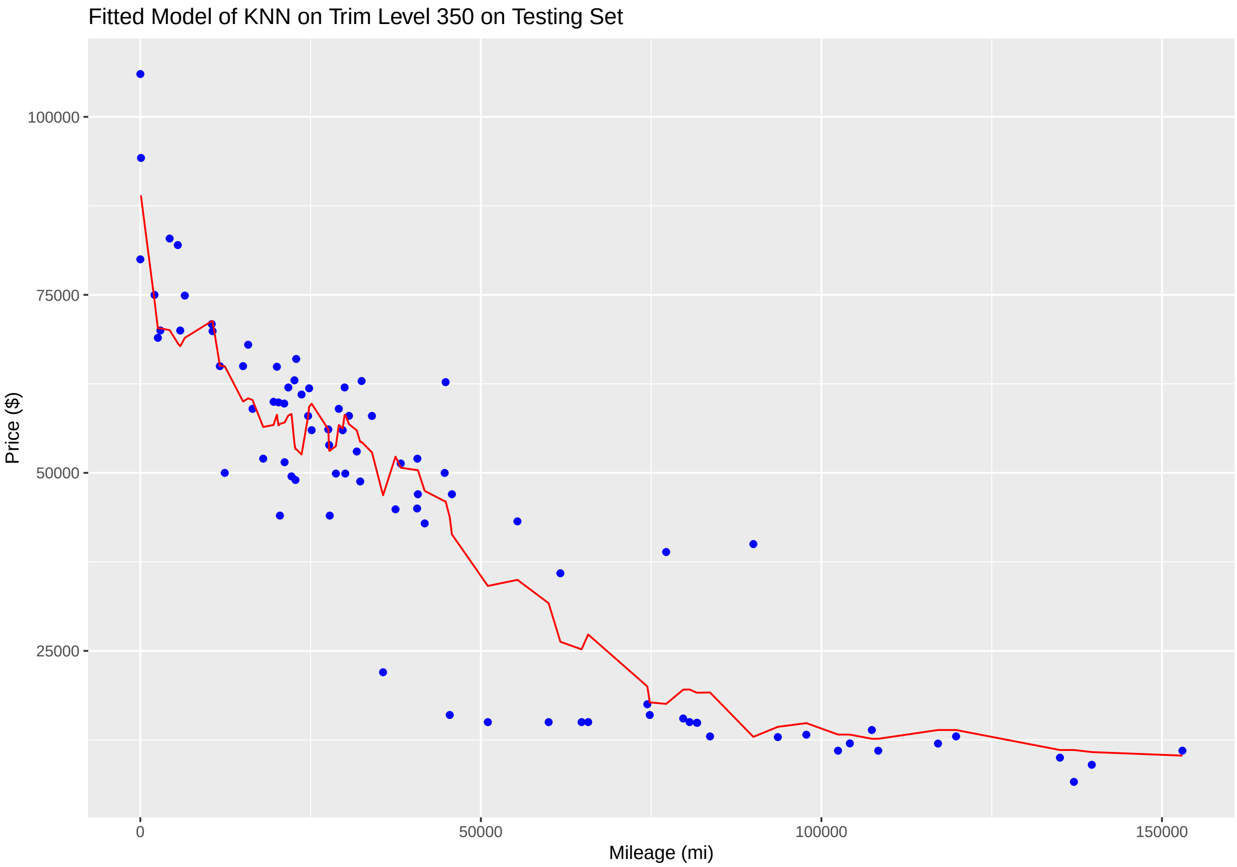


To identify which k values are better than others for the KNN regression model, we used the root mean square error ($RMSE$) to quantify the quality of the fit between the actual values and the predicted values. $RMSE$ measures the differences between values predicted by a hypothetical model and the observed values. The formula for $RMSE$ is $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2}$.

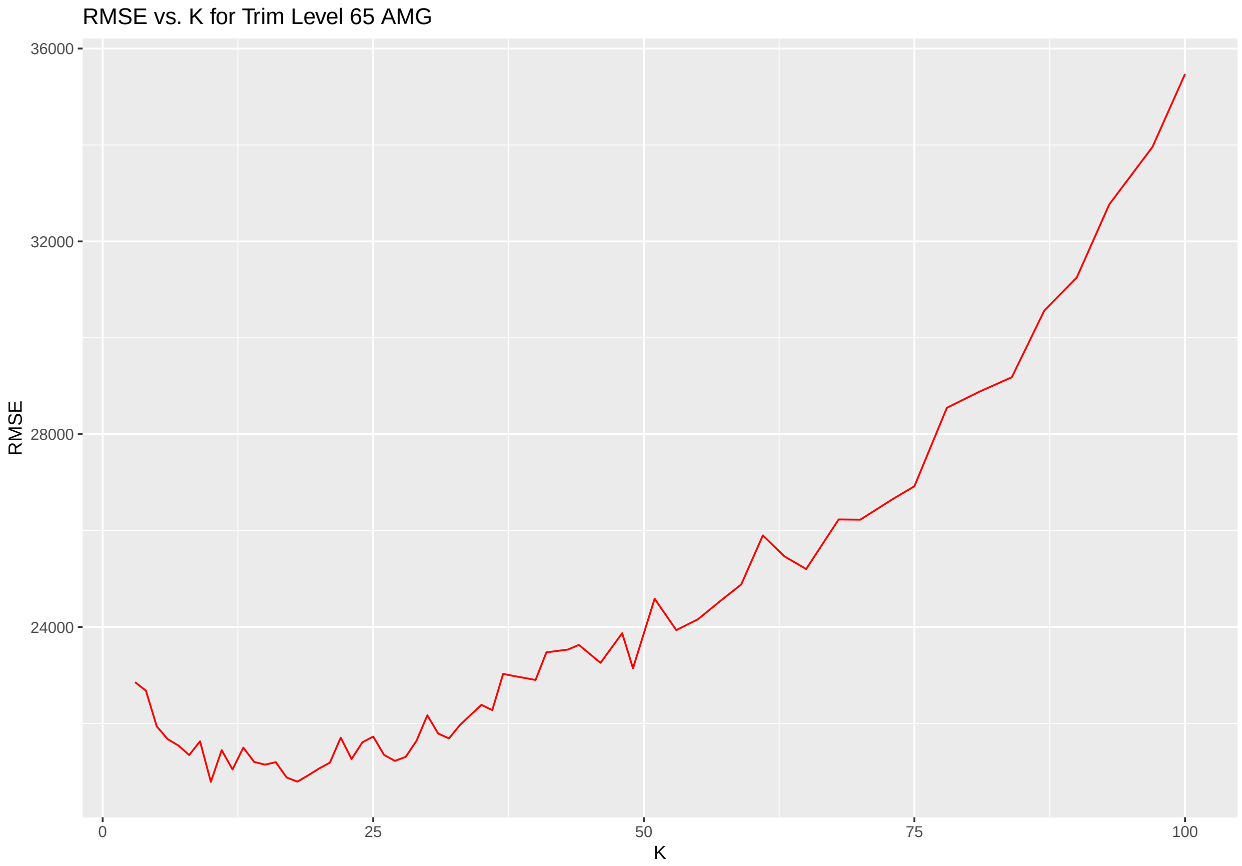
For our KNN regression models, we followed the standardized procedure of splitting our original data into training and testing sets where 80% of the data was used in training and the remaining 20% of the data was used in testing. For each of the several values of k that we tested ranging between 3 and 100, we ran 200 train/test splits and computed the mean $RMSE$. We ran 200 train/test splits for each value of k in order to reduce the variation of each $RMSE$ value computed. In order to reduce the Monte Carlo variability, we ran each split numerous times because running a single train/test split could result in vastly different values for the $RMSE$ (and inherently our choice of the optimal k value) for each run. Using these $RMSE$ values, we were able to select the optimal value of k by choosing the k value with the smallest $RMSE$ value.



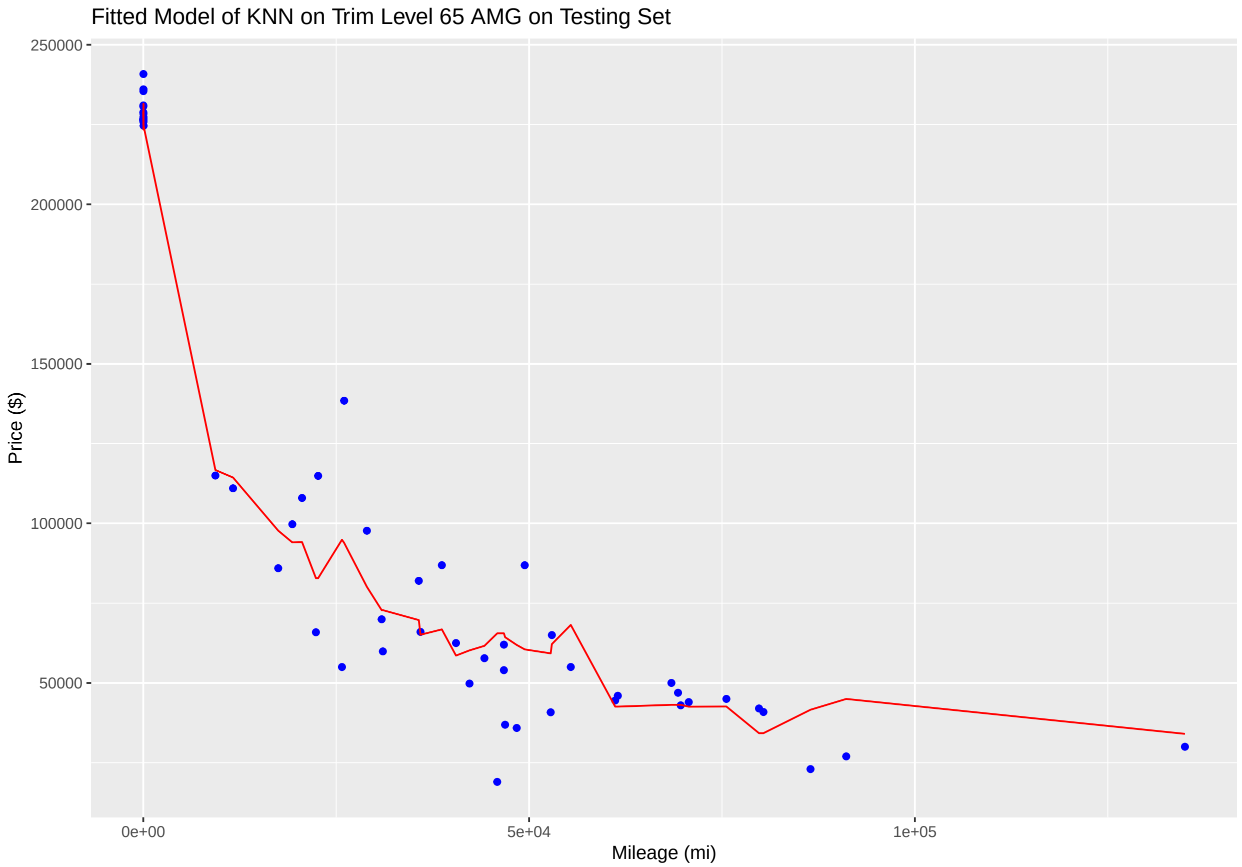
Fitting a KNN regression model to predict price from mileage for Mercedes S class vehicles with trim 350 over various values of k, we have found that the optimal k value is 12 with a $RMSE$ of approximately 9951.19. In other words, $k = 12$ was the value at which the difference between the actual price and the predicted price was minimal for trim 350. In the plot below, you can see that $k = 12$ has the lowest $RMSE$ value out of the k values that were tested.



Fitting this KNN regression model on a 20% testing set for the optimal $k = 12$, we can see that this model provides a reasonably accurate price prediction for the 350 trim of Mercedes S class vehicles. From this fitted model, we can see that there is some thrashing in the price range of $\$25,000$ to $\$35,000$, as mentioned earlier. Since the data points are more spread out in this area, it is reasonable that some predictions in this range do not follow a noticeable pattern as they do in prices outside of this range. This is because the nearest neighbors for data points in this price range are “farther” than that of data points in the other price ranges.



Fitting a KNN regression model to predict price from mileage for Mercedes S class vehicles with trim 65 AMG over various values of k, we have found that the optimal k value is 10 with a $RMSE$ of approximately 20788.64. In other words, $k = 10$ was the value at which the difference between the actual price and the predicted price was minimal for trim 65 AMG. In the plot below, you can see that $k = 10$ has the lowest $RMSE$ value out of the k values that were tested.



Fitting this KNN regression model on a 20% testing set for the optimal $k = 10$, we can see that this model also provides a reasonably accurate price prediction for the 65 AMG of Mercedes S class vehicles. This particular testing set has some outliers with very low mileage and high price as well as some outliers with very high mileage and low price. This KNN model still shows an accurate fitted line because the neighbors to these outliers help shape the regression line.

Conclusion

How does the optimal KNN regression model for trim 350 compare to that for trim 65 AMG?

After running KNN regression on both Mercedes S class vehicles with trim 350 and trim 65 AMG, we can see that the optimal k value is larger for trim 350 ($12 > 10$). This indicates that the trim 350 data set regression model needed more nearest neighbors compared to trim 65 AMG. This is logical because the trim 350 data values are more dense over the whole range but they are concentrated in two clearly defined price ranges. Since the trim 65 AMG data values are less dense and more evenly spread out throughout the range of prices, the KNN regression model required fewer nearest neighbors to make an accurate prediction for price. However, trim 65 AMG has a larger $RMSE$ value ($20788.64 > 9951.19$). This indicates that the regression model for trim 350 is a better fit than the regression model for trim 65 AMG. This is reasonable because the regression model for trim 350 utilized more nearest neighbors than the regression model for 65 AMG, resulting in predictions that were closer to the actual prices. Since the 65 AMG trim is a type of higher-end sports car while the 350 trim is a more middle-range luxury sedan, it is rational that the 65 AMG trim data is more spread out and more sparse compared to the 350 trim, which is more dense and contains more overall data points.

Saratoga house prices

Another type of price prediction that is very useful and important in our society is predicting house prices. Many housing retail and leasing companies use regression and other types of models to predict the estimated market rate for houses based on various features. For example, we could use number of rooms, utility status, fuel system, average income of residents in neighboring houses, etc. to help predict the market rate price for a house.

The data that we analyzed for this case includes more than 1500 houses and several of their features, including number of rooms, age of the house, living area, and percentage of neighboring residents with a college degree, among other important factors (shown below).

```
##    price lotSize age landValue livingArea pctCollege bedrooms fireplaces
## 1 132500    0.09  42     50000        906         35        2          1
## 2 181115    0.92   0     22300       1953         51        3          0
## 3 109000    0.19 133      7300       1944         51        4          1
## 4 155000    0.41  13     18700       1944         51        3          1
## 5  86060    0.11   0     15000        840         51        2          0
## 6 120000    0.68  31     14000       1152         22        4          1
##   bathrooms rooms         heating     fuel             sewer waterfront
## 1       1.0     5        electric electric            septic         No
## 2       2.5     6 hot water/steam      gas            septic         No
## 3       1.0     8 hot water/steam      gas public/commercial         No
## 4       1.5     5         hot air      gas            septic         No
## 5       1.0     3         hot air      gas public/commercial         No
## 6       1.0     8         hot air      gas            septic         No
##   newConstruction centralAir
## 1              No         No
## 2              No         No
## 3              No         No
## 4              No         No
## 5             Yes        Yes
## 6              No         No
```

For more information on this data set, here is a summary of its data:

```
##      price           lotSize             age           landValue     
##  Min.   :  5000   Min.   : 0.0000   Min.   :  0.00   Min.   :   200  
##  1st Qu.:145000   1st Qu.: 0.1700   1st Qu.: 13.00   1st Qu.: 15100  
##  Median :189900   Median : 0.3700   Median : 19.00   Median : 25000  
##  Mean   :211967   Mean   : 0.5002   Mean   : 27.92   Mean   : 34557  
##  3rd Qu.:259000   3rd Qu.: 0.5400   3rd Qu.: 34.00   3rd Qu.: 40200  
##  Max.   :775000   Max.   :12.2000   Max.   :225.00   Max.   :412600  
##    livingArea     pctCollege       bedrooms       fireplaces       bathrooms  
##  Min.   : 616   Min.   :20.00   Min.   :1.000   Min.   :0.0000   Min.   :0.0  
##  1st Qu.:1300   1st Qu.:52.00   1st Qu.:3.000   1st Qu.:0.0000   1st Qu.:1.5  
##  Median :1634   Median :57.00   Median :3.000   Median :1.0000   Median :2.0  
##  Mean   :1755   Mean   :55.57   Mean   :3.155   Mean   :0.6019   Mean   :1.9  
##  3rd Qu.:2138   3rd Qu.:64.00   3rd Qu.:4.000   3rd Qu.:1.0000   3rd Qu.:2.5  
##  Max.   :5228   Max.   :82.00   Max.   :7.000   Max.   :4.0000   Max.   :4.5  
##      rooms                   heating            fuel     
##  Min.   : 2.000   hot air        :1121   gas     :1197  
##  1st Qu.: 5.000   hot water/steam: 302   electric: 315  
##  Median : 7.000   electric       : 305   oil     : 216  
##  Mean   : 7.042
##  3rd Qu.: 8.250
##  Max.   :12.000
##                sewer      waterfront newConstruction centralAir
##  septic           : 503   Yes:  15   Yes:  81        Yes: 635
##  public/commercial:1213   No :1713   No :1647        No :1093
##  none             :  12
##
##
##
```

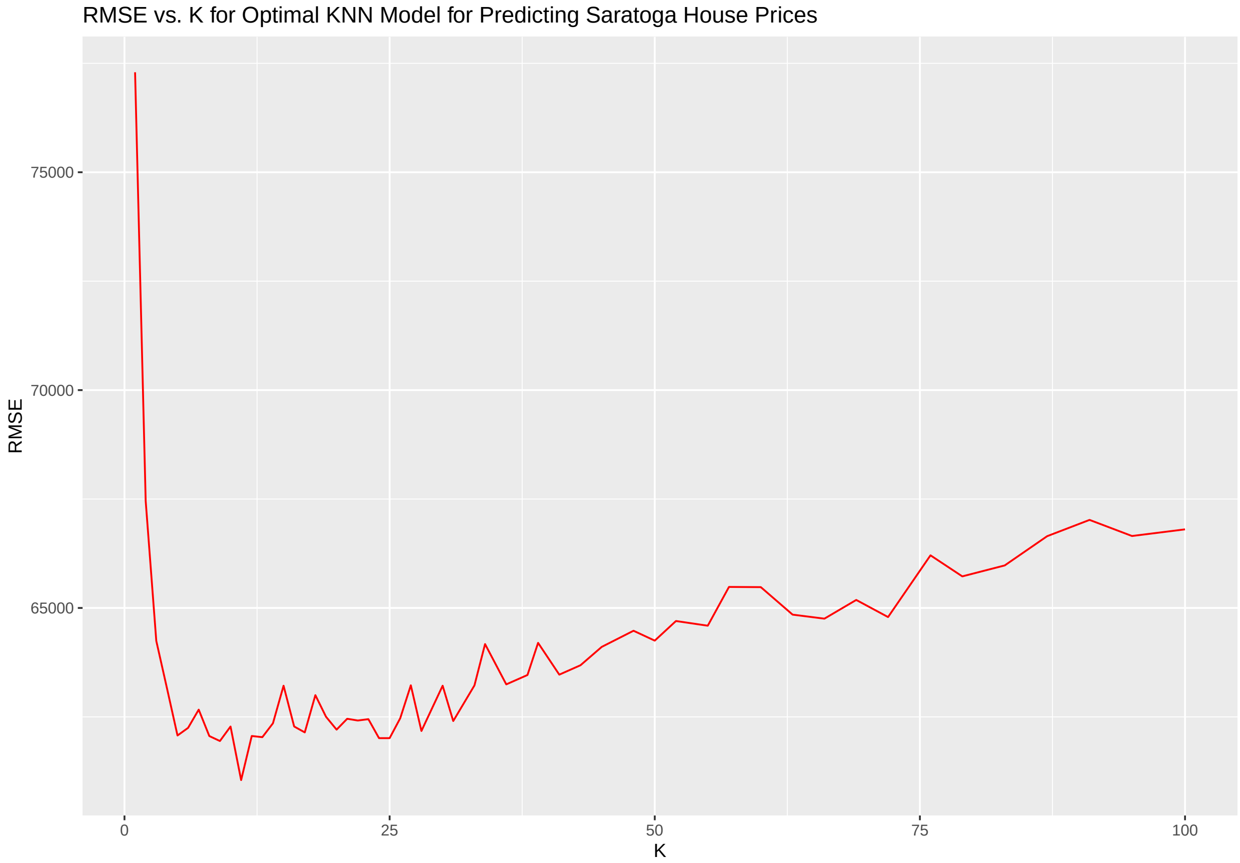
To identify which variables and interactions produced better price predictions, we used the root mean square error ($RMSE$) to quantify the quality of the fit between the actual values and the predicted values for each model (similar to the previous problem). For reference, the formula we used to calculate the $RMSE$ is $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2}$.

For our KNN regression models, we followed the same standardized procedure of splitting our original data into training and testing sets where 80% of the data was used in training and the remaining 20% of the data was used in testing. For each of the several values of k that we tested ranging between 3 and 100, we ran 200 train/test splits and computed the mean $RMSE$. We ran 200 train/test splits for each value of k in order to reduce the variation of each $RMSE$ value computed. In order to reduce the Monte Carlo variability, we ran each split numerous times because running a single train/test split could result in vastly different values for the $RMSE$ (and inherently our choice of the optimal k value) for each run. Using these $RMSE$ values, we were able to select the optimal value of k by choosing the k value with the smallest $RMSE$ value.

To help us select factors and interactions that produced more accurate regression models, we first looked at the p-value of the first 3 basic linear models that were provided. We observed that the first linear regression model showed that number of bedrooms and bathrooms were factors that facilitated predicting the house market rate price. We also noticed that the second linear regression model showed that the number of rooms and the percentage of residents in the neighborhood with a college degree were significant factors that made the prediction more accurate. We then took into account that the third linear regression model showed that the living area of the house (in sqaure feet) was a primary factor in price prediction. In addition to this, we also analyzed the Akaike information criterion (AIC) values of each of the regression models to help us decide which ones were useful to consider. The AIC is an estimator of out-of-sample prediction error and thereby relative quality of statistical models for a given set of data by dealing with the risks of overfitting and underfitting.

After performing this analysis to select some of our variables of interest, we then tried various combinations of these variables and interactions between them in a systematic manner. We kept track of the $RMSE$ values produced from the linear regression of each of these combinations of variables and interatcions. Our final linear regression model took all pairwise interactions between living area of the house, number of bedrooms, number of bathrooms, and total number of rooms in addition to all pairwise interactions between heating system type and central air conditioning as well as the individual variables of fuel system type and percentage of residents in the neighborhood with a college degree.

In other words, we found it plausible that the effect of the number of bedrooms and bathrooms on price depended on the living area of the house and vice versa. The intuition behind this idea is that houses with more sqaure feet in living area typically have more bedrooms and bathrooms. This also shows that the living area of the house predicts price based on the number of bedrooms and bathrooms. For example, if two houses A and B have an equal square feet of living area but A has fewer bedrooms and bathrooms than B, then it is reasonable that the market rate price for house A is less than that of house B. Similarly, we found that the interactions between heating system and air conditioning predicted the market rate price for a house because the presence of one and not the other typically resulted in a lower market rate than if both were present.



For our KNN model, we used the standardized process to identify the optimal value of k for the variables and implicit interactions that we used by selecting the k value with the smallest $RMSE$. Since the KNN model can account for interactions between any variables used, we simpy had to only include the variables that we were interested in using for prediction (the same unqiue variables that we used in our linear regression model). Above, we have plotted the $RMSE$ value for each k value that we tested.

```
## Mean RMSE of Linear Regression Model #1: 77630.45
```

```
## Mean RMSE of Linear Regression Model #2: 66727.39
```

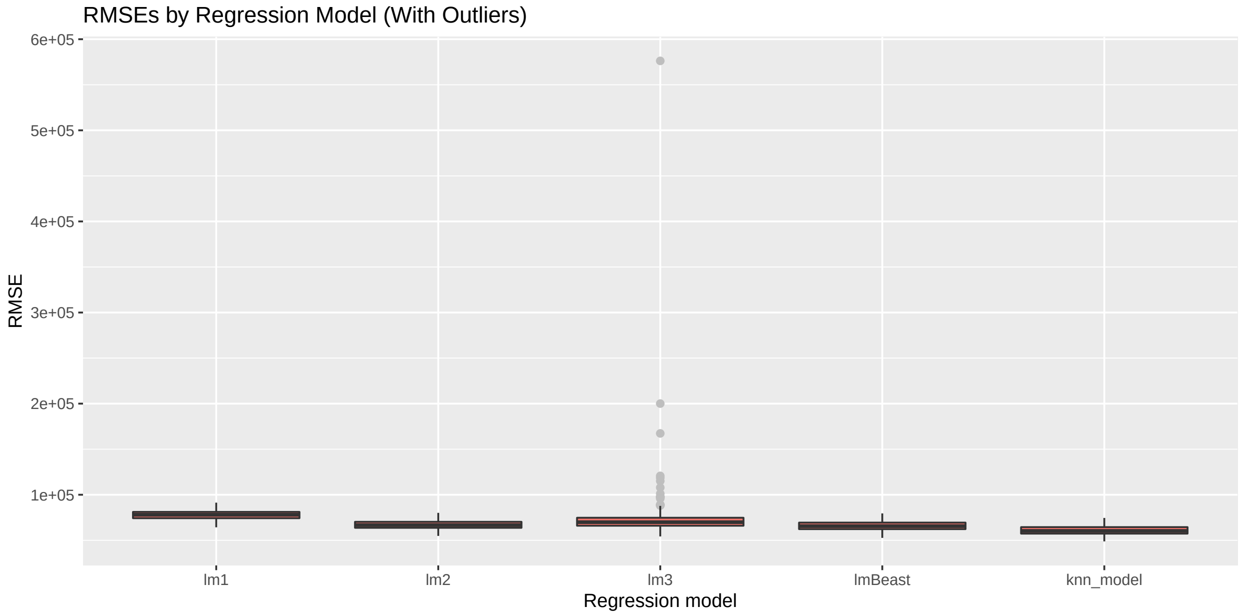
```
## Mean RMSE of Linear Regression Model #3: 75349.97
```

```
## Mean RMSE of our Linear Regression Model: 65716.94
```

```
## Mean RMSE of our KNN Regression Model: 61049.51
```

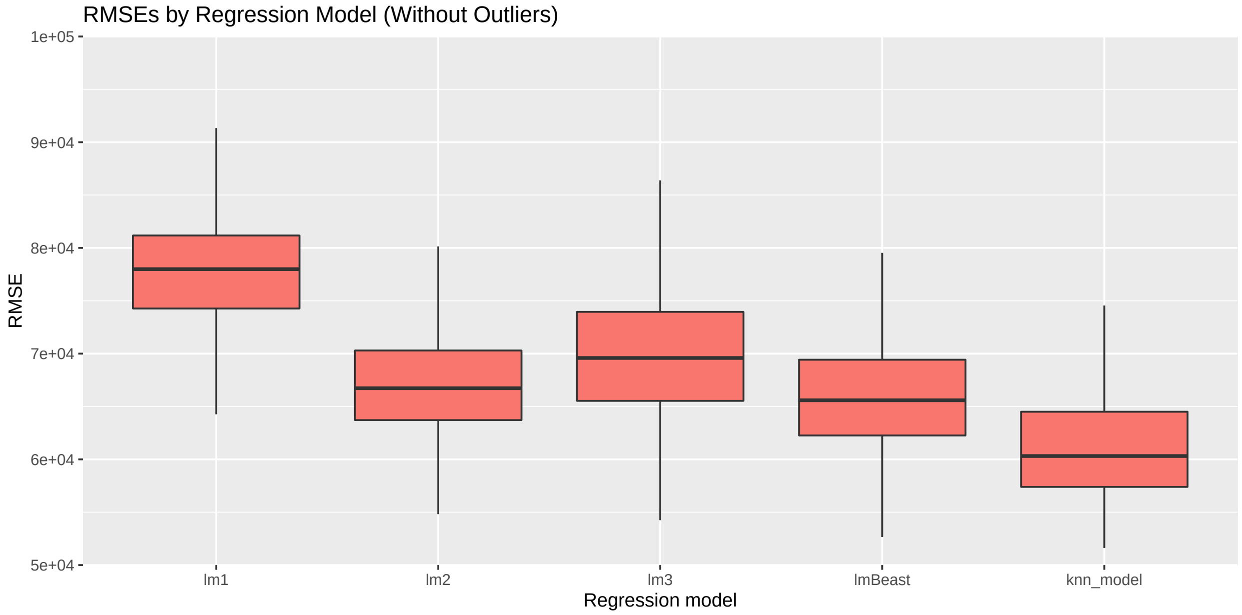
The individual mean $RMSE$ values, which we computed in order to allow us to compare models, for each of the regression models that we analyzed are shown above. We can see that Linear Regression Model #2 has a better fit than Linear Regression Models #1 and #3. We also observe that our Linear Regression Model produced a better fit than Linear Regression Model #2 by approximately 1000 units. Our final KNN Regression Model was the best fit of all of the models that we tested. Our KNN Regression Model produced a better fit than Linear Regression Model #2 by approximately 5000 units.

```
## No id variables; using all as measure variables
```



```
## No id variables; using all as measure variables
```

```
## Warning: Removed 9 rows containing non-finite values (stat_boxplot).
```



The above boxplots show more detailed information regarding the $RMSE$ values for each of the regression models. We have provided the boxplots with and without outliers in order to highlight the extremity of the outliers for Linear Regression Model #3 as well as the noticeable decrease in average $RMSE$ for our Linear Regression Model and even more so for our KNN Regression Model.

Conclusion

Which variables and interactions accurately predict market values for properties?

From the perspective of a local taxing authority looking to form predicted market values for properties in order to know how much to tax them, we know that we would like to pick the best combination of variables and interactions with the best regression model in order to make the most accurate price predictions. From our data analysis, we have shown that the best predictions for housing market rates came from the using a KNN regression model taking into account all variables (and implicitly the interactions between these variables) except for heating system type, fuel system type, sewer, waterfront, new construction, and central air conditioning. Specifically, we performed various types of data anlysis on the provided variables in the data set in order to understand that the most significant variables to help in the prediction of market rate prices for houses include the living area of a house, the number of bedrooms, bathrooms, and total rooms, the air conditioning system, and the percent of residents in the neighborhood with a college degree. The interactions that we found between living area in square feet and the number of bedrooms and bathrooms is also important to take into account, as explained above.

Predicting when articles go viral

Another interesting prediction that we can make is whether or not a given article will go viral. With the rise of social media in our society, this is a insightful prediction to make because it will provide us with a better understanding of what kind of information people are more willing to consume and circulate within their social groups.

The data that we analyzed for this case includes approximately 40,000 online articles published by Mashable during 2013 and 2014. The target variable in this data set is *shares*, which is the number of times that an articles is shared online. For this case, we consider an article to be 'viral' if it has more than $1,400$ shares. The other variables are article-level features: length of the headline, length of the article, and how positive or negative the "sentiment" of the article was, among many other specific technical features.

```
##                                                               url n_tokens_title
## 1   http://mashable.com/2013/01/07/amazon-instant-video-browser/             12
## 2    http://mashable.com/2013/01/07/ap-samsung-sponsored-tweets/              9
## 3 http://mashable.com/2013/01/07/apple-40-billion-app-downloads/              9
## 4       http://mashable.com/2013/01/07/astronaut-notre-dame-bcs/              9
## 5               http://mashable.com/2013/01/07/att-u-verse-apps/             13
## 6               http://mashable.com/2013/01/07/beewi-smart-toys/             10
##   n_tokens_content num_hrefs num_self_hrefs num_imgs num_videos
## 1              219         4              2        1          0
## 2              255         3              1        1          0
## 3              211         3              1        1          0
## 4              531         9              0        1          0
## 5             1072        19             19       20          0
## 6              370         2              2        0          0
##   average_token_length num_keywords data_channel_is_lifestyle
## 1             4.680365            5                         0
## 2             4.913725            4                         0
## 3             4.393365            6                         0
## 4             4.404896            7                         0
## 5             4.682836            7                         0
## 6             4.359459            9                         0
##   data_channel_is_entertainment data_channel_is_bus data_channel_is_socmed
## 1                             1                   0                      0
## 2                             0                   1                      0
## 3                             0                   1                      0
## 4                             1                   0                      0
## 5                             0                   0                      0
## 6                             0                   0                      0
##   data_channel_is_tech data_channel_is_world self_reference_min_shares
## 1                    0                     0                       496
## 2                    0                     0                         0
## 3                    0                     0                       918
## 4                    0                     0                         0
## 5                    1                     0                       545
## 6                    1                     0                      8500
##   self_reference_max_shares self_reference_avg_sharess weekday_is_monday
## 1                       496                    496.000                 1
## 2                         0                      0.000                 1
## 3                       918                    918.000                 1
## 4                         0                      0.000                 1
## 5                     16000                   3151.158                 1
## 6                      8500                   8500.000                 1
##   weekday_is_tuesday weekday_is_wednesday weekday_is_thursday weekday_is_friday
## 1                  0                    0                   0                 0
## 2                  0                    0                   0                 0
## 3                  0                    0                   0                 0
## 4                  0                    0                   0                 0
## 5                  0                    0                   0                 0
## 6                  0                    0                   0                 0
##   weekday_is_saturday weekday_is_sunday is_weekend global_rate_positive_words
## 1                   0                 0          0                 0.04566210
## 2                   0                 0          0                 0.04313725
## 3                   0                 0          0                 0.05687204
## 4                   0                 0          0                 0.04143126
## 5                   0                 0          0                 0.07462687
## 6                   0                 0          0                 0.02972973
##   global_rate_negative_words avg_positive_polarity min_positive_polarity
## 1                0.013698630             0.3786364            0.10000000
## 2                0.015686275             0.2869146            0.03333333
## 3                0.009478673             0.4958333            0.10000000
## 4                0.020715631             0.3859652            0.13636364
## 5                0.012126866             0.4111274            0.03333333
## 6                0.027027027             0.3506100            0.13636364
##   max_positive_polarity avg_negative_polarity min_negative_polarity
## 1                   0.7            -0.3500000                -0.600
## 2                   0.7            -0.1187500                -0.125
## 3                   1.0            -0.4666667                -0.800
## 4                   0.8            -0.3696970                -0.600
## 5                   1.0            -0.2201923                -0.500
## 6                   0.6            -0.1950000                -0.400
##   max_negative_polarity title_subjectivity title_sentiment_polarity
## 1            -0.2000000          0.5000000               -0.1875000
## 2            -0.1000000          0.0000000                0.0000000
## 3            -0.1333333          0.0000000                0.0000000
## 4            -0.1666667          0.0000000                0.0000000
## 5            -0.0500000          0.4545455                0.1363636
## 6            -0.1000000          0.6428571                0.2142857
##   abs_title_sentiment_polarity shares
## 1                    0.1875000    593
## 2                    0.0000000    711
## 3                    0.0000000   1500
## 4                    0.0000000   1200
## 5                    0.1363636    505
## 6                    0.2142857    855
```

For more information on this data set, here is a summary of its data:

```
##                                                               url
##  http://mashable.com/2013/01/07/amazon-instant-video-browser/  :    1
##  http://mashable.com/2013/01/07/ap-samsung-sponsored-tweets/   :    1
##  http://mashable.com/2013/01/07/apple-40-billion-app-downloads/:    1
##  http://mashable.com/2013/01/07/astronaut-notre-dame-bcs/      :    1
##  http://mashable.com/2013/01/07/att-u-verse-apps/              :    1
##  http://mashable.com/2013/01/07/beewi-smart-toys/              :    1
##  (Other)                                                       :39638
##  n_tokens_title n_tokens_content   num_hrefs      num_self_hrefs
##  Min.   : 2.0   Min.   :   0.0   Min.   :  0.00   Min.   :  0.000
##  1st Qu.: 9.0   1st Qu.: 246.0   1st Qu.:  4.00   1st Qu.:  1.000
##  Median :10.0   Median : 409.0   Median :  8.00   Median :  3.000
##  Mean   :10.4   Mean   : 546.5   Mean   : 10.88   Mean   :  3.294
##  3rd Qu.:12.0   3rd Qu.: 716.0   3rd Qu.: 14.00   3rd Qu.:  4.000
##  Max.   :23.0   Max.   :8474.0   Max.   :304.00   Max.   :116.000
##
##     num_imgs         num_videos    average_token_length  num_keywords
##  Min.   :  0.000   Min.   : 0.00   Min.   :0.000        Min.   : 1.000
##  1st Qu.:  1.000   1st Qu.: 0.00   1st Qu.:4.478        1st Qu.: 6.000
##  Median :  1.000   Median : 0.00   Median :4.664        Median : 7.000
##  Mean   :  4.544   Mean   : 1.25   Mean   :4.548        Mean   : 7.224
##  3rd Qu.:  4.000   3rd Qu.: 1.00   3rd Qu.:4.855        3rd Qu.: 9.000
##  Max.   :128.000   Max.   :91.00   Max.   :8.042        Max.   :10.000
##
##  data_channel_is_lifestyle data_channel_is_entertainment data_channel_is_bus
##  Min.   :0.00000           Min.   :0.000                 Min.   :0.0000
##  1st Qu.:0.00000           1st Qu.:0.000                 1st Qu.:0.0000
##  Median :0.00000           Median :0.000                 Median :0.0000
##  Mean   :0.05295           Mean   :0.178                 Mean   :0.1579
##  3rd Qu.:0.00000           3rd Qu.:0.000                 3rd Qu.:0.0000
##  Max.   :1.00000           Max.   :1.000                 Max.   :1.0000
##
##  data_channel_is_socmed data_channel_is_tech data_channel_is_world
##  Min.   :0.0000         Min.   :0.0000       Min.   :0.0000
##  1st Qu.:0.0000         1st Qu.:0.0000       1st Qu.:0.0000
##  Median :0.0000         Median :0.0000       Median :0.0000
##  Mean   :0.0586         Mean   :0.1853       Mean   :0.2126
##  3rd Qu.:0.0000         3rd Qu.:0.0000       3rd Qu.:0.0000
##  Max.   :1.0000         Max.   :1.0000       Max.   :1.0000
##
##  self_reference_min_shares self_reference_max_shares self_reference_avg_sharess
##  Min.   :     0            Min.   :     0            Min.   :     0.0
##  1st Qu.:   639            1st Qu.:  1100            1st Qu.:   981.2
##  Median :  1200            Median :  2800            Median :  2200.0
##  Mean   :  3999            Mean   : 10329            Mean   :  6401.7
##  3rd Qu.:  2600            3rd Qu.:  8000            3rd Qu.:  5200.0
##  Max.   :843300            Max.   :843300            Max.   :843300.0
##
##  weekday_is_monday weekday_is_tuesday weekday_is_wednesday weekday_is_thursday
##  Min.   :0.000     Min.   :0.0000     Min.   :0.0000       Min.   :0.0000
##  1st Qu.:0.000     1st Qu.:0.0000     1st Qu.:0.0000       1st Qu.:0.0000
##  Median :0.000     Median :0.0000     Median :0.0000       Median :0.0000
##  Mean   :0.168     Mean   :0.1864     Mean   :0.1875       Mean   :0.1833
##  3rd Qu.:0.000     3rd Qu.:0.0000     3rd Qu.:0.0000       3rd Qu.:0.0000
##  Max.   :1.000     Max.   :1.0000     Max.   :1.0000       Max.   :1.0000
##
##  weekday_is_friday weekday_is_saturday weekday_is_sunday   is_weekend
##  Min.   :0.0000    Min.   :0.00000     Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000    1st Qu.:0.00000     1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000    Median :0.00000     Median :0.00000   Median :0.0000
##  Mean   :0.1438    Mean   :0.06188     Mean   :0.06904   Mean   :0.1309
##  3rd Qu.:0.0000    3rd Qu.:0.00000     3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000    Max.   :1.00000     Max.   :1.00000   Max.   :1.0000
##
##  global_rate_positive_words global_rate_negative_words avg_positive_polarity
##  Min.   :0.00000            Min.   :0.000000           Min.   :0.0000
##  1st Qu.:0.02838            1st Qu.:0.009615           1st Qu.:0.3062
##  Median :0.03902            Median :0.015337           Median :0.3588
##  Mean   :0.03962            Mean   :0.016612           Mean   :0.3538
##  3rd Qu.:0.05028            3rd Qu.:0.021739           3rd Qu.:0.4114
##  Max.   :0.15549            Max.   :0.184932           Max.   :1.0000
##
##  min_positive_polarity max_positive_polarity avg_negative_polarity
##  Min.   :0.00000       Min.   :0.0000        Min.   :-1.0000
##  1st Qu.:0.05000       1st Qu.:0.6000        1st Qu.:-0.3284
##  Median :0.10000       Median :0.8000        Median :-0.2533
##  Mean   :0.09545       Mean   :0.7567        Mean   :-0.2595
##  3rd Qu.:0.10000       3rd Qu.:1.0000        3rd Qu.:-0.1869
##  Max.   :1.00000       Max.   :1.0000        Max.   : 0.0000
##
##  min_negative_polarity max_negative_polarity title_subjectivity
##  Min.   :-1.0000       Min.   :-1.0000       Min.   :0.0000
##  1st Qu.:-0.7000       1st Qu.:-0.1250       1st Qu.:0.0000
##  Median :-0.5000       Median :-0.1000       Median :0.1500
##  Mean   :-0.5219       Mean   :-0.1075       Mean   :0.2824
##  3rd Qu.:-0.3000       3rd Qu.:-0.0500       3rd Qu.:0.5000
##  Max.   : 0.0000       Max.   : 0.0000       Max.   :1.0000
##
##  title_sentiment_polarity abs_title_sentiment_polarity     shares
##  Min.   :-1.00000         Min.   :0.0000               Min.   :     1
##  1st Qu.: 0.00000         1st Qu.:0.0000               1st Qu.:   946
##  Median : 0.00000         Median :0.0000               Median :  1400
##  Mean   : 0.07143         Mean   :0.1561               Mean   :  3395
##  3rd Qu.: 0.15000         3rd Qu.:0.2500               3rd Qu.:  2800
##  Max.   : 1.00000         Max.   :1.0000               Max.   :843300
##
```

First, we will compute a baseline KNN model that will predict ‘not viral’ for every instance. Since we will be producing more useful regression models, it is important to have a baseline regression model to compare with to indicate whether the regression models we develop are truly useful. The confusion matrix and the accuracy rate, error rate, true positive rate, and false positive rate are shown below for the baseline KNN regression model.

```
##              Predicted

## Actual        Viral        Not viral

##   Viral       0         3997

##   Not viral   0         3932

## The accuracy rate for the baseline KNN regression model is 0.4959011

## The error rate for the baseline KNN regression model is 0.5040989

## The true positive rate for the baseline KNN regression model is 0

## The false positive rate for the baseline KNN regression model is 0
```

To identify which variables and interactions produced better price predictions for our optimal KNN regression model, we used the root mean square error ($RMSE$) to quantify the quality of the fit between the actual values and the predicted values for each model (similar to the previous problem). For reference, the formula we used to calculate the $RMSE$ is $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2}$.

For our KNN regression models, we followed the same standardized procedure of splitting our original data into training and testing sets where 80% of the data was used in training and the remaining 20% of the data was used in testing. For each of the several values of k that we tested ranging between 3 and 100, we ran 100 train/test splits and computed the mean $RMSE$. We ran 100 train/test splits for each value of k in order to reduce the variation of each $RMSE$ value computed. In order to reduce the Monte Carlo variability, we ran each split numerous times because running a single train/test split could result in vastly different values for the $RMSE$ (and inherently our choice of the optimal k value) for each run. Using these $RMSE$ values, we were able to select the optimal value of k by choosing the k value with the smallest $RMSE$ value.

For this KNN regression model, we intended to initially predict the number of shares for each instance in the testing set, and then threshold the predicted value as ‘viral’ or ‘not viral’. In order to develop a more useful KNN regression model, we took a similar approach as the previous exercise where we verified the usefulness of specific variables in linear regression models with their p-values. This helped us identify several variables that could help predict shares, including the length of the title and the article, the sentiment of the article, the day of the week it was posted on, etc. The confusion matrix and the accuracy rate, error rate, true positive rate, and false positive rate are shown below for our optimal KNN regression model.

```
##              Predicted

## Actual        Viral        Not viral

##   Viral       3262         724

##   Not viral   2975         968
## The accuracy rate for the KNN regression model is 0.5334847
```

```
## The error rate for the KNN regression model is 0.4665153
```

```
## The true positive rate for the KNN regression model is 0.8183643
```

```
## The false positive rate for the KNN regression model is 0.7545016
```

Now, to further test if we could improve our predictions, we used a logistic regression to first threshold the instances in the data set as 'viral' or 'not viral', and then predict whether the testing set instances were 'viral' or 'not viral'. We did this by selecting predicted values above 0.5 as 'viral' and 'not viral' otherwise. To minimize the variability in the results, we averaged the results over numerous train/test splits. The confusion matrix and the accuracy rate, error rate, true positive rate, and false positive rate are shown below for our logistic regression model.

```
##             Predicted

## Actual        Viral        Not viral

##   Viral       2842         1193.7

##   Not viral   2068.7        1823.6
```

```
## The accuracy rate for the logistic regression model is 0.5884965
```

```
## The error rate for the logistic regression model is 0.4115035
```

```
## The true positive rate for the logistic regression model is 0.7042149
```

```
## The false positive rate for the logistic regression model is 0.5314852
```

Conclusion

Which approach performs better: regress first and threshold second, or threshold first and regress/classify second?

Since the accuracy rate of our optimal KNN regression model is greater than the accuracy rate of the baseline KNN model, we know that our KNN model performs better than the baseline model by predicting shares. Since the accuracy rate of our logistic regression model is greater than the accuracy rates of both the optimal KNN regression model and the baseline KNN regression model, we claim that we have developed an even better regression model for predicting 'viral' or 'not viral'.

From fitting these regression models, we can see that the approach of thresholding first and regressing/classifying second works better for this data set to predict whether an article will be 'viral' or 'not viral'. This is because the accuracy rate for our logistic regression model is greater than the accuracy rate of our KNN regression model. We believe that this is due to the fact that classifying as 'viral' and 'not viral' is a smaller scope to deal with, so the factors that the logistic regression takes into account will be more accurate and will be a better predictor of the viral status of an article.