SDS323: Exercise 2

Max Kutschinski (mwk556)

03/13/2020

KNN practice

Introduction

The goal of this exercise is to predict the price of different Mercedes S-Class vehicles using K-nearest-neighbors. For websites such as Cars.com and Truecar, providing accurate pricing information to consumers has traditionally been a notoroiously difficult case when it comes to the Mercedes S-Class. This is largely due to the fact that there is a wide range of sub-models that are all labeled S-Class. For this particular exercise, the focus will be on two different trim levels (350 and 65 AMG), as well as one feature variable (mileage).

Data and Methods

The dataset contains data on over 29,000 Mercedes S-Class vehicles. Since we are only interested in looking at two different trim levels, two subsets will be extracted from the full dataset, and the 350's and the 65 AMG's will be treated as two separate datasets. The target variable is price, and the feature to be analyzed per trim level is mileage.

Table 1 shows some basic summary statistics of the mileage on the car for each of the two trim levels. The datset consisting of all the S-Class 350 models contains 416 observations, while the S-Class 65 AMG dataset contains 292 oberservations.

Table 1: Summary stats (mileage)

trim	count	min	max	mean	median
350	416	6	173,000	42,926	29,998
65 AMG	292	1	146,975	33,700	28,803

To analyze the data, the same procedure will be followed for each trim level:

1. The data will be split into a training and a testing set (80/20) in order to avoid overfitting.
2. K-nearest-neighbors will be run for many different values of K, starting at K=2. For each value of K, the model will be fit to the training set and predictions will be made on the test set.
3. The out-of-sample root mean-squared error (RMSE) will be calculated for each value of K.

When measuring out of sample performance, there exists random variation due to the particular choice of data points that end up in the train/test split. Thus, both KNN modes will be compared using their average out-of-sample RMSE and K over multiple different train/test splits. To be specific, each K value will be evaluated over 100 different train/test splits.

Results

(i) Mercedes S-Class 350

Figure 1 and 2 highlight the results from analyzing the S-Class 350 models. Figure 1 demonstrates the RMSE for all possible K values. The graph also highlights the optimal value of K, that is the value that minimizes the RMSE. According to this plot the optimal K value is at K equal to 10.

Figure 1

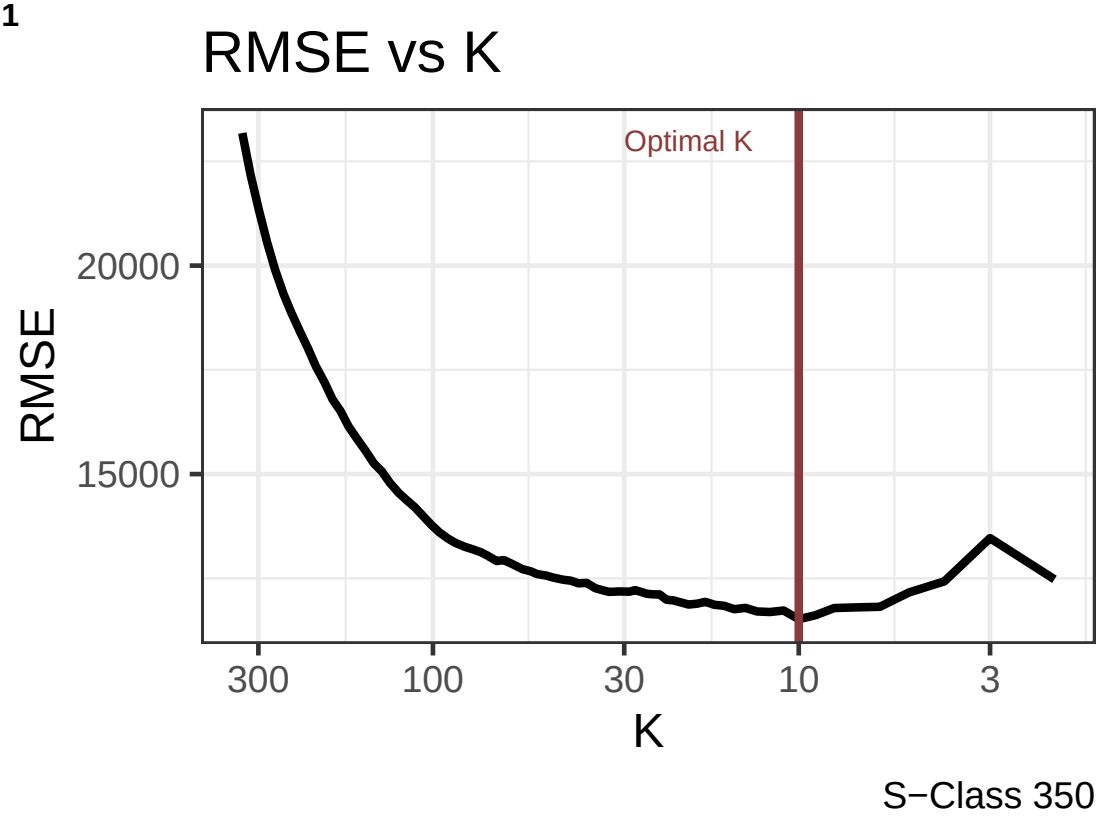
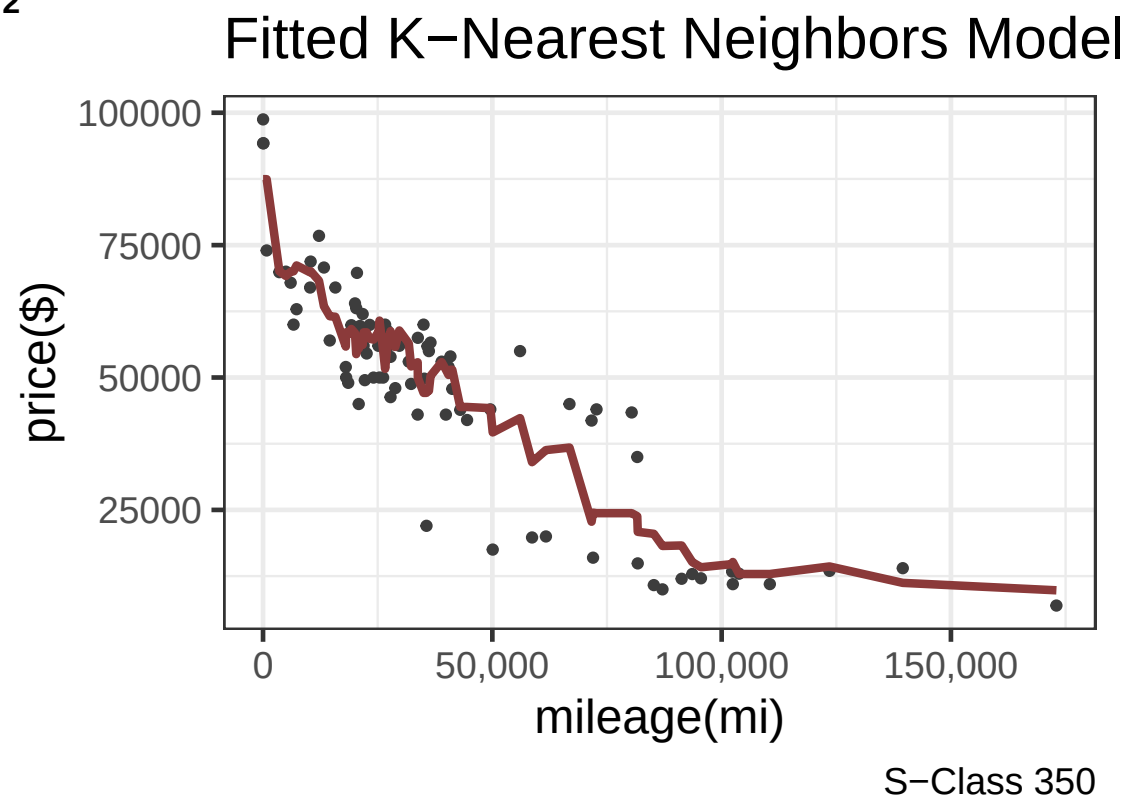


Figure 2 is an extension of Figure 1, by plotting the k-nearest neighbors model that utilizes the optimal K value found in Figure 1.

Figure 2



(ii) Mercedes S-Class 65 AMG

Figure 3 and 4 highlight the results from analyzing the S-Class 65 AMG models. Here again, Figure 3 demonstrates the RMSE for all possible K values. The graph also highlights the optimal value of K, that is the value that minimizes the RMSE. According to this plot, the optimal K value is at K equal to 34.

Figure 3

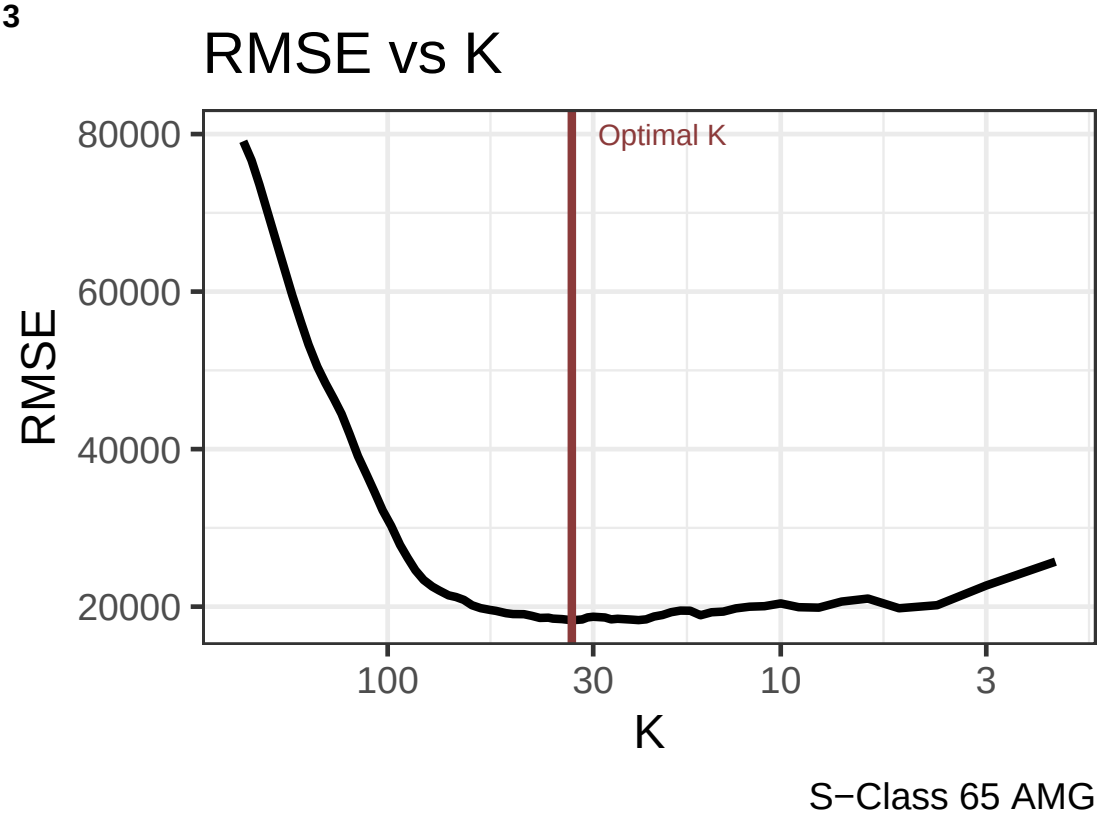
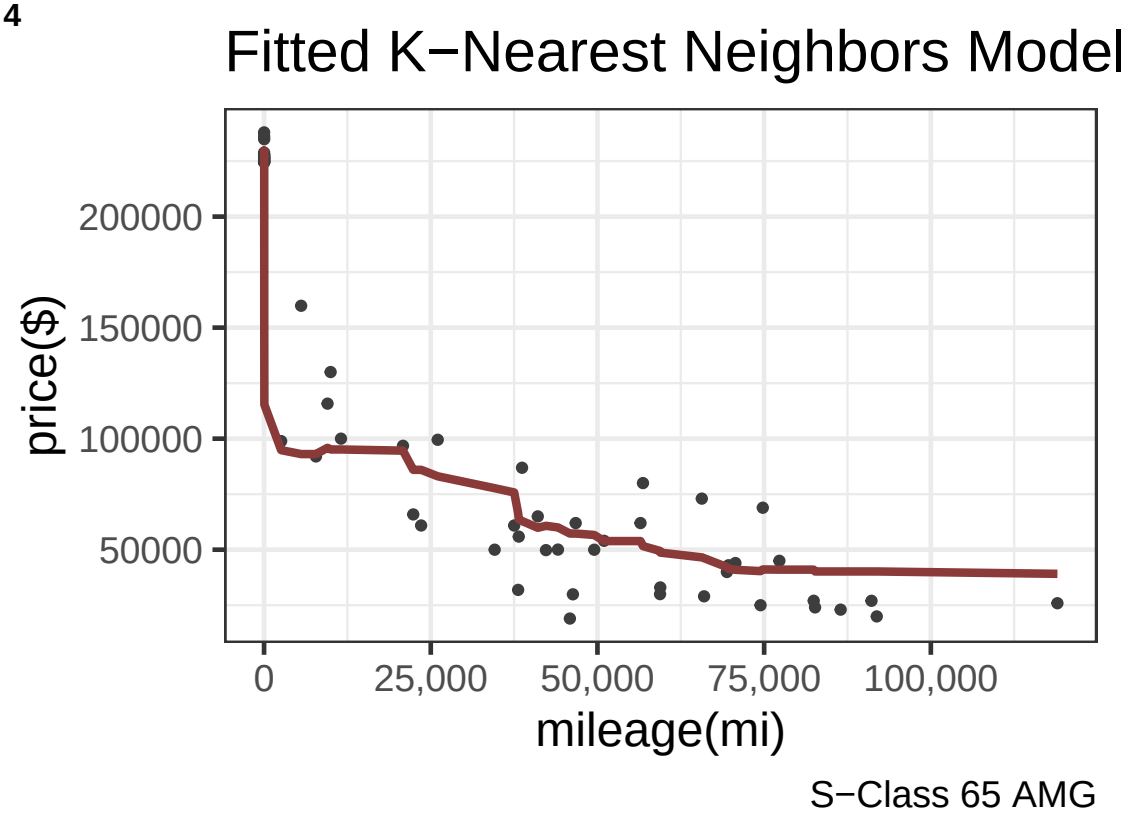


Figure 4 is an extension of Figure 3, by plotting the k-nearest neighbors model that utilizes the optimal K value found in Figure 3.

Figure 4



Conclusions

Overall, it seems like the KNN model worked better for the 350 trim level, since the RMSE is a lot lower in comparison (see Figure 1 and 3).

When comparing trim levels, there also appears to be a difference in the optimal value for K. The S-Class 350 model had an optimal K value of 10, while the S-Class 65 AMG model had an optimal K value of 34. This difference in K values ultimately comes down to the bias variance tradeoff. There exists a tradeoff between a model's ability to minimize bias and variance. A larger K value adds complexity to the model, resulting in higher variance, but lower bias. The opposite is true for low K values. Thus, the reason why one model has a higher k value is because that model is more sensitive to random noise.

Saratoga house prices

Introduction

The goal of this exercise is to build a model for predicting house prices in Saratoga, NY. The task is to create a linear model that outperforms the old model discussed in class, as well as the construction of a K-nearest-neighbors (KNN) model with the same features. In addition, the two models developed in this paper will be compared to see which one performs better.

Data and Methods

The dataset contains 1728 observations of 16 variables. These variables are summarized in Table 2, where the price is the response variable we are seeking to predict.

Table 2: List of Variables

x
price
lotSize
age
landValue
livingArea
pctCollege
bedrooms
fireplaces
bathrooms
rooms
heating
fuel
sewer
waterfront
newConstruction
centralAir

Table 3 summarizes the features used in the two models, which aim to outperform the old model. The older model included all variables except for sewer, waterfront, landValue, and newConstruction. The linear and the KNN model will be compared using the same features, with the exception of interactions, which are omitted in the K-nearest-neighbors model (KNN is sufficiently adaptable to find them). The reason behind deciding to include two interactions in the linear model is that there seems to be a dependent relationship between some variables. For example, in the real world the number of bedrooms is usually not much different from the number of bathrooms. A studio apartment with five bathrooms seems somewhat unlikely. Thus, it makes sense to include an interaction variable that accounts for this effect.

Table 3: Features of the Linear Model

x
age
lotSize
landValue
livingArea
pctCollege
bedrooms
bathrooms
rooms
bedrooms:bathrooms
livingArea:newConstruction

When measuring out of sample performance, there exists random variation due to the particular choice of data points that end up in the train/test split. Thus, both the linear- and the KNN model will be compared using their average out-of-sample RMSE over multiple different train/test splits.

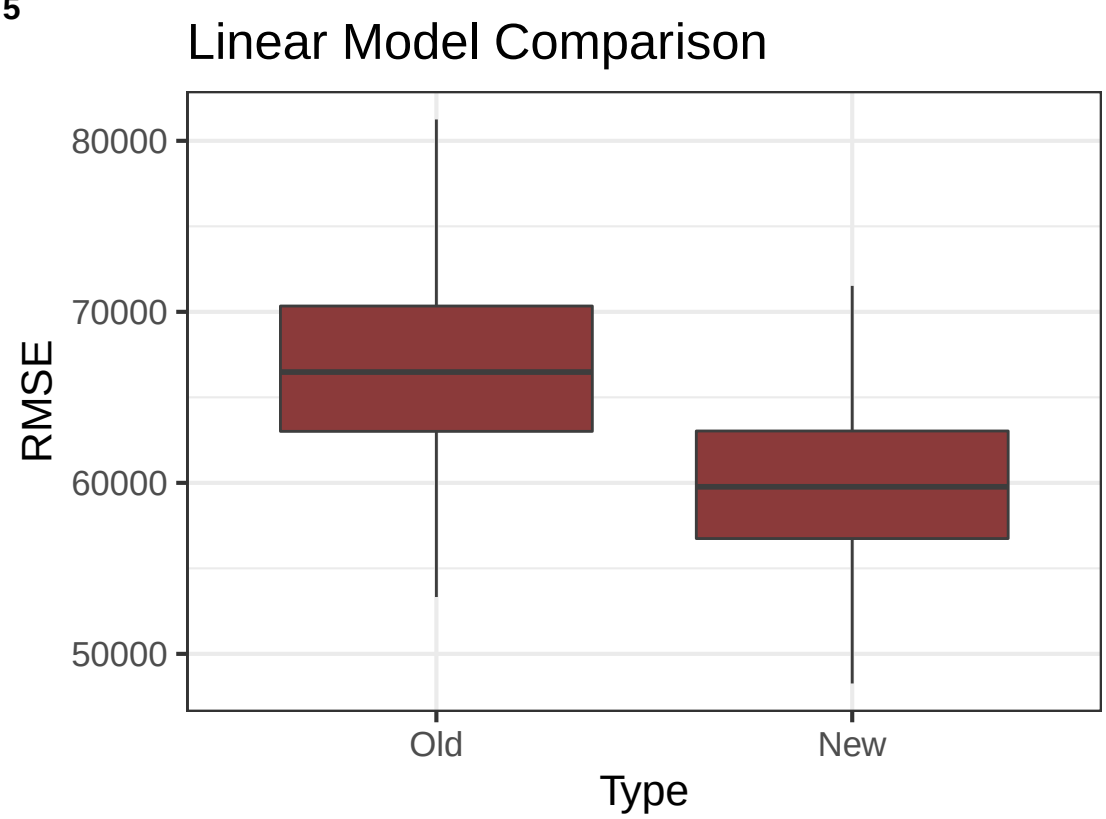
Before running the KNN regression, it is also important to standardize the variables: A one unit change lotSize (sqft) is not equal to a one unit change in the number of bedrooms. To capture this effect, we will use z scores to measure relative effects.

Results

(i) Linear Model

The new linear model outperforms the old model by a significant margin, as can be seen in Figure 5. When averaging over 500 different train/test splits, the old model has a RMSE of around 66,637, while the new model has a RMSE of approximately 59,957.

Figure 5



According to the new model, the strongest drivers of house prices are landValue, livingArea, bathrooms and the interaction between livingArea and NewConstruction. Table 5 summarizes the new linear model and highlights the significance levels of individual features. The strong drivers listed above all have extremely low p values of below 0.001.

Table 5: Properties of Regression Coefficients

	price		
Predictors	*Estimates*	*CI*	*p*
(Intercept)	20980.31	-18836.26 – 60796.87	0.301
age	-220.29	-338.17 – -102.42	**<0.001**
lotSize	4344.06	-498.91 – 9187.03	0.079
landValue	1.00	0.90 – 1.10	**<0.001**
livingArea	50.89	39.64 – 62.14	**<0.001**
pctCollege	-186.90	-508.50 – 134.69	0.254
bedrooms	-6700.21	-18165.60 – 4765.18	0.252
bathrooms	29635.59	10341.34 – 48929.85	**0.003**
rooms	3335.25	1202.64 – 5467.87	**0.002**
bedrooms * bathrooms	-1321.85	-6857.16 – 4213.45	0.640
livingArea : newConstructionNo	18.72	12.44 – 24.99	**<0.001**
Observations	1382		

(ii) KNN Model

Figure 6 deals with the KNN model and describes the RMSE for many different values of K. In this model the value of K that minimizes the RMSE is 9.

Figure 6

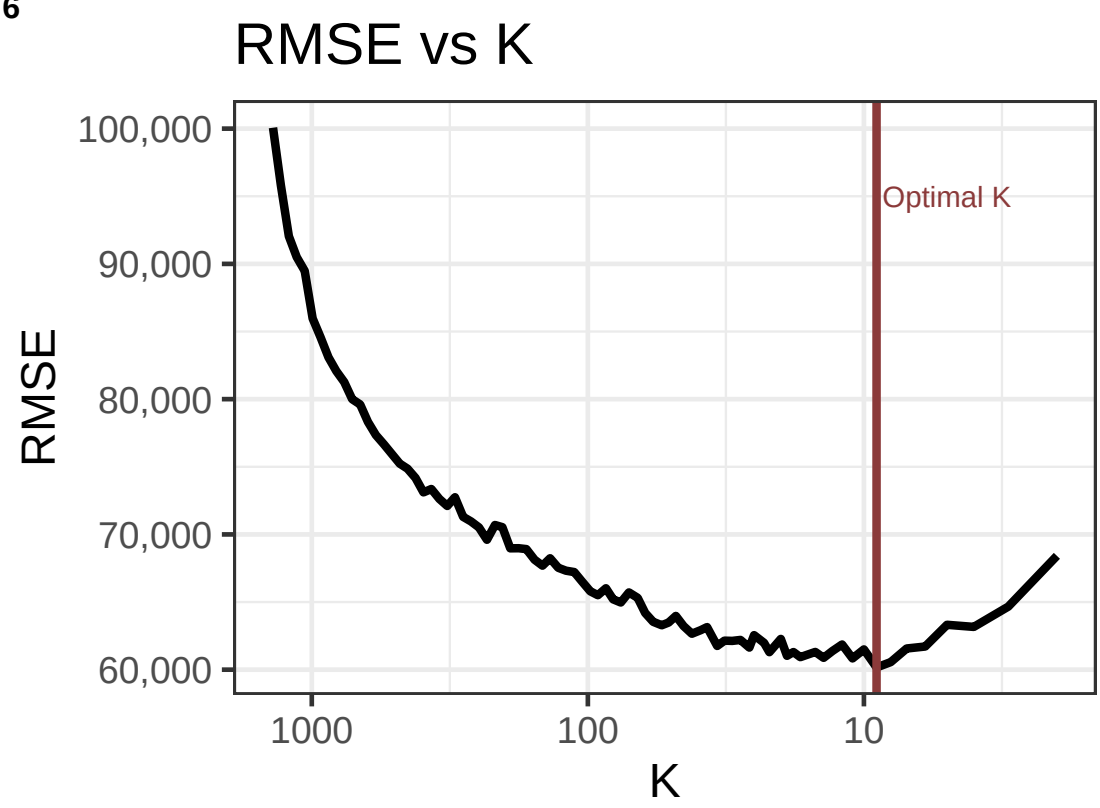
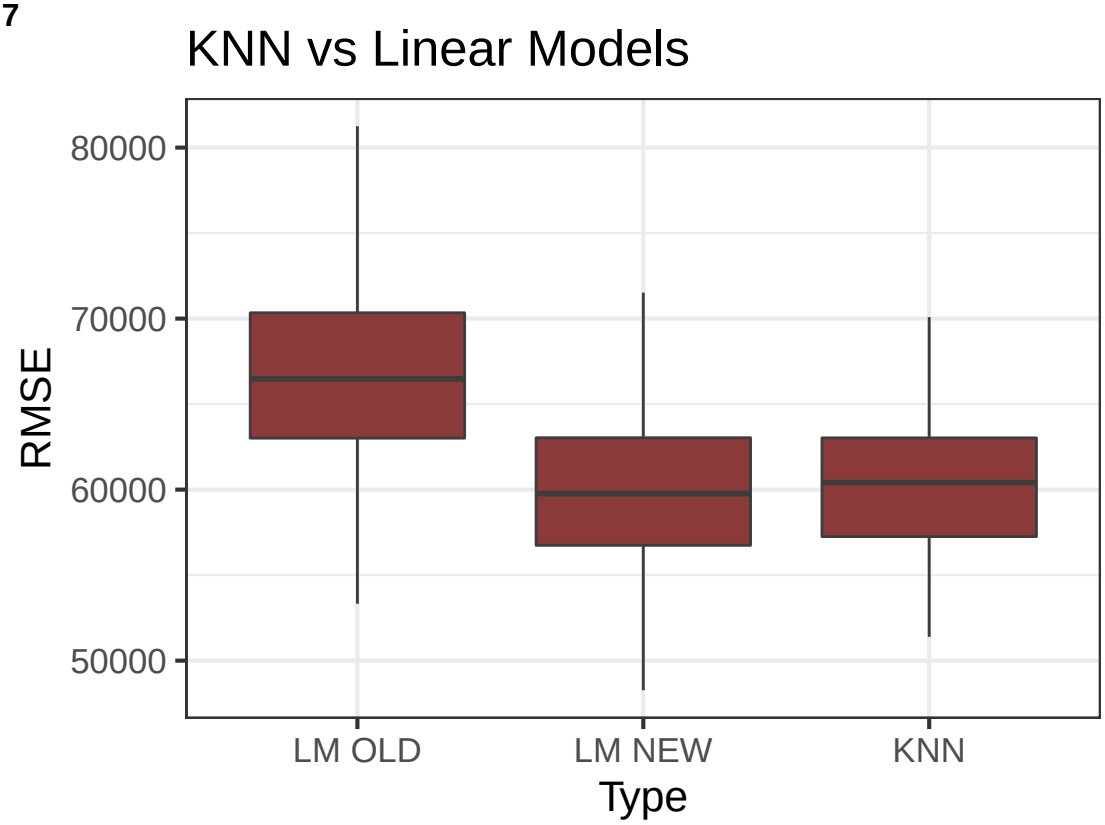


Figure 7 captures and compares the RMSE of all three models. It is apparent that both newly developed models perform better than the old model. Among the new models, the linear model seems to perform slightly better.

Figure 7



Conclusions

Overall, these results indicate that feature selection can make a big difference. In fact, it seems like variables such as land value, living area, and new construction are particularly strong drivers of house prices, and are therefore crucial to consider when thinking about price-modeling strategies. Based on these results, the linear model will perform better than a KNN in the future, and constitutes therefore a viable option for predicting house prices.

Predicting when articles go viral

Introduction

The goal of this exercise is to predict when articles go viral. The company Mashable considers an article viral if it was shared more than 1400 times. They are interested in knowing if there's anything they can learn about how to improve an article's chance of reaching this threshold. (E.g by telling people to write shorter headlines, snarkier articles, or whatever.)

Data and Methods

The provided dataset contains data on 39,797 online articles published by Mashable during 2013 and 2014. It encompasses 38 variables, where target variable is shares. Other variables that can be used for the analysis are article-level features and are summarized in Table 4.

Table 4: List of Feature Variables

Variable	Definition
n_tokens_title	Number of words in the title
n_tokens_content	Number of words in the content
num_hrefs	Number of links
num_self_hrefs	Number of links to other articles published by Mashable
num_imgs	Number of images
num_videos	Number of videos
average_token_length	Average length of the words in the content
num_keywords	Number of keywords in the metadata
data_channel_is_lifestyle	Is data channel "Lifestyle"?
data_channel_is_entertainment	Is data channel "Entertainment"?
data_channel_is_bus	Is data channel "Business"?
data_channel_is_socmed	Is data channel "Social Media"?
data_channel_is_tech	Is data channel "Tech"?
data_channel_is_world	Is data channel "World"?
self_reference_min_shares	Min. shares of referenced articles in Mashable
self_reference_max_shares	Max. shares of referenced articles in Mashable
self_reference_avg_sharess	Avg. shares of referenced articles in Mashable
weekday_is_Monday	Was the article published on a Monday?
weekday_is_Tuesday	Was the article published on a Tuesday?
weekday_is_Wednesday	Was the article published on a Wednesday?
weekday_is_Thursday	Was the article published on a Thursday?
weekday_is_Friday	Was the article published on a Friday?
weekday_is_Saturday	Was the article published on a Saturday?
weekday_is_Sunday	Was the article published on a Sunday?
global_rate_positive_words	Rate of positive words in the content
global_rate_negative_words	Rate of negative words in the content
avg_positive_polarity	Avg. polarity of positive words
min_positive_polarity	Min. polarity of positive words
max_positive_polarity	Max. polarity of positive words
avg_negative_polarity	Avg. polarity of negative words
min_negative_polarity	Min. polarity of negative words
max_negative_polarity	Max. polarity of negative words
title_subjectivity	Title polarity
title_sentiment_polarity	Title subjectivity

The task of this exercise is to approach the problem by using two different methods: regression, and classification. Before diving into these techniques, however, a null model will be established that can be used as a reference point to compare different models. A quick look at the dataset reveals that there are slightly more articles that did not go viral (Table 6).

Table 6: Data exploration

	Count
Not Viral	20082
Viral	19562

Thus, the constructed null model will always predict "not viral", which results in an error rate below 50 percent when applied to the entire data set.

In both modeling approaches, the model will be built on all available features. When approaching this problem from a regression standpoint, it is important to threshold the model's predictions. That is,

- if predicted shares exceed 1400, predict the article as "viral"
- if predicted shares are 1400 or lower, predict the article as "not viral"

The regression model of choice will be stepwise linear regression with forward selection, in order to make predictions about viral status. In this model, all features will be used as a basis for forward selection. However, interdependence was not accounted for (no interactions or squared terms).

On the other hand, when approaching this problem from a standpoint of classification, viral status will be directly predicted as a target variable. Thus 'viral' will first be defined as a new variable, which is binary and takes values of 1 if shares exceed 1400, and 0 if not.

For this part, the Naive Bayes Classifier will be the model of choice.

Finally, both approaches will be compared using the average error rate, true positive rate, and false positive rate over multiple train/test splits. To compare equal quantities between the models, a total of 5 splits will be used. Note that the accuracy of the results will go up with the number of splits.

Results

(i) Regression

Table 7 captures the confusion matrix of the linear model, while Table 8 captures the confusion matrix of the null model. The error rates of each model are summarized in Table 9. Notice that the linear model actually performed worse than the Null model, since it's error rate is slightly higher in comparison.

Table 7: Confusion Matrix (Linear Model)

	yhat	
	0	1
y=0	331	19798
y=1	134	19382

Table 8: Confusion Matrix (Null Model)

	yhat	
	0	1
y=0	20082	0
y=1	19562	0

Table 9: Error Rates (Regression)

	Linear	Null
Error_rate (%)	50.28	49.34
TPR (%)	99.31	0.00
FPR (%)	98.36	0.00

(ii) Classification

Table 10 captures the confusion matrix for the naive bayes model, while Table 11 captures the confusion matrix for the null model. Table 12 sumamrizes the error rates for both models. Note that in this case, the naive bayes model outperforms the null model.

Table 10: Confusion Matrix (Naive Bayes)

	yhat	
	0	1
y=0	15575	4535
y=1	13156	6379

Table 11: Confusion Matrix (Null Model)

	yhat	
	0	1
y=0	20082	0
y=1	19562	0

The naive bayes model shows an absolute improvement of around 4.72 % and a lift over the null model of around 1.09.

Table 12: Error Rates (Classification)

	Naive Bayes	Null
Error_rate (%)	44.62	49.34
TPR (%)	32.65	0.00
FPR (%)	22.55	0.00

Conclusions

Overall, it is safe to say that the approach of threshold first and regress/classify second (classification) performed better than the regress first and threshold second approach. One likely explanation for this is that regression is more sensitive to variation in the data. In a classification model, all shares above 1400 are valued the same (viral), whereas a regression model imposes a ranking between different viral and non viral articles before it classifies them. This implies that features of articles with a lot shares, say 3000, will be weighted more significant in the regression than features of articles with 1500 shares,for example. The naive bayes model outperformed the linear-, as well as the null model. For further research, it would make sense to look at other models such as KNN or logistic regression to check if they can outperform the proposed naive bayes model.