BMW Used Car Sales Recommender System

Yukang Xu

1. Overview

BMW is a German multinational corporation which produces luxury vehicles and motorcycles. Over the past few years, BMW has built up many famous model lines such as M series, X series. To help BMW salesperson better understand clients' needs and create a platform for potential buyers to search their dream cars, I would use BMW used car sales data to build an adaptive recommender system. Taken into consideration of specific requests make by customers (e.g., Mileage, Model year, etc.), our system would list out the information of top 10 suitable used cars.

To target suitable recommendations, I would apply K-Prototype clustering methods at first then use classification model to assign new request to the most relevant cluster. Calculated correlations with each member in this cluster would rank used car information from being the most fitting to the least.

The main purpose of this recommender system to navigate used car selection process either for clients or salesperson. In general, our recommender system could be used in two cases. When salespeople introduce products to customers via phone or in store, they could input their understanding of clients' needs into our system. Then salespeople could use the output of system to kick off their introduction. The second circumstance is when our potential buyers are looking for advice on their next car on the website. Typing in all information required by buyers, our recommender system is able to put up its personalized suggestions. All in all, BMW sales team could use the help of this system to boost the quality of customer services and user experience.

2. Descriptive Analysis

Before we start, I would like to introduce the dataset I would be using. All the variables and corresponding descriptions are attached below.

Variable (Data type)	Description
Model (Object)	i.e., 1 Series, M2
Year (Int)	The year of car produced
Price (Int)	Selling price
Transmission (Object)	i.e., Manual, Automatic
Mileage (Int)	How many miles a car lasts
fuelType (Object)	i.e., Diesel, Petrol
Tax (Int)	Taxes required to pay when purchasing
Mpg (Float)	Milage per gallon
engineSize (Float)	Total volume of the cylinders in the engine

Variable description

Let's check out numeric main statistics of numeric variables. In the light of median and mean shown in table 1, average used car is produced in 2017, sold by $20k, running 20k miles with 2 engines equipped. Great news!! From table 2, we have **no missing value** to deal with.

	year	price	mileage	tax	mpg	engineSize
count	10781.000000	10781.000000	10781.000000	10781.000000	10781.000000	10781.000000
mean	2017.078935	22733.408867	25496.986550	131.702068	56.399035	2.167767
std	2.349038	11415.528189	25143.192559	61.510755	31.336958	0.552054
min	1996.000000	1200.000000	1.000000	0.000000	5.500000	0.000000
25%	2016.000000	14950.000000	5529.000000	135.000000	45.600000	2.000000
50%	2017.000000	20462.000000	18347.000000	145.000000	53.300000	2.000000
75%	2019.000000	27940.000000	38206.000000	145.000000	62.800000	2.000000
max	2020.000000	123456.000000	214000.000000	580.000000	470.800000	6.600000

Main statistics summary of dataset

Table 1

```
RangeIndex: 10781 entries, 0 to 10780
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   model         10781 non-null  object
 1   year          10781 non-null  int64
 2   price         10781 non-null  int64
 3   transmission  10781 non-null  object
 4   mileage       10781 non-null  int64
 5   fuelType      10781 non-null  object
 6   tax           10781 non-null  int64
 7   mpg           10781 non-null  float64
 8   engineSize    10781 non-null  float64
dtypes: float64(2), int64(4), object(3)
```

Missing value summary

Table 2

Figure 1 below shows us the positive relationship between year and price while figure 2 and 3 finds no relationship between year and mpg (Mileage per gallon) or price and tax.

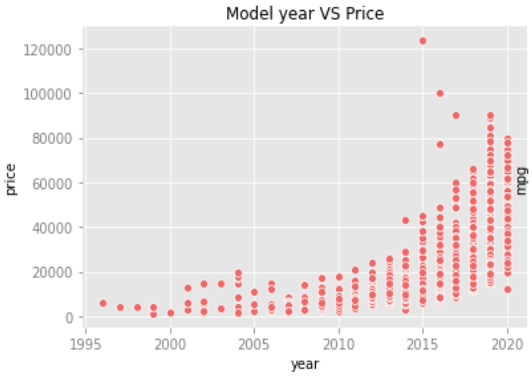


Figure 1

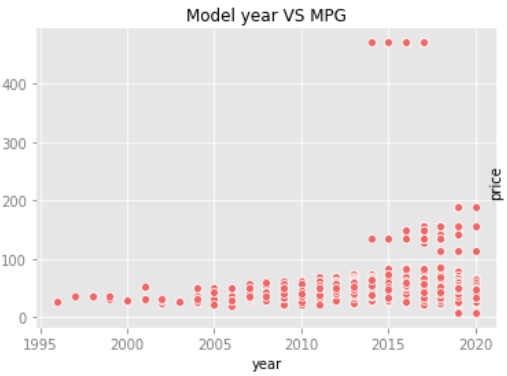


Figure 2

Figure 3

Figure 4 shown below, BWM 1 series and 3 series appear to be the most popular car model. Semi-auto cars are the bestselling products from figure 5. Figure 6 is a little counterintuitive to American customs, cars with diesel engines are the favorites choices of buyers.

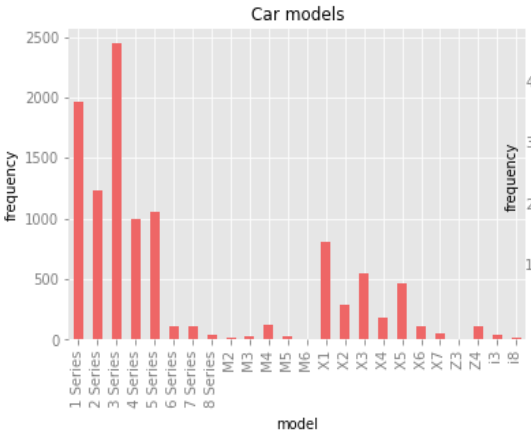


Figure 4

Figure 5

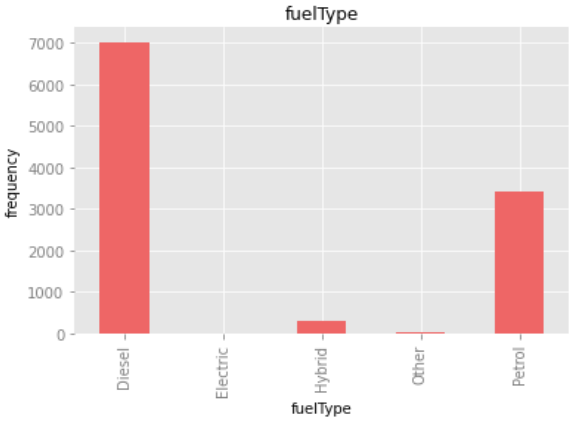


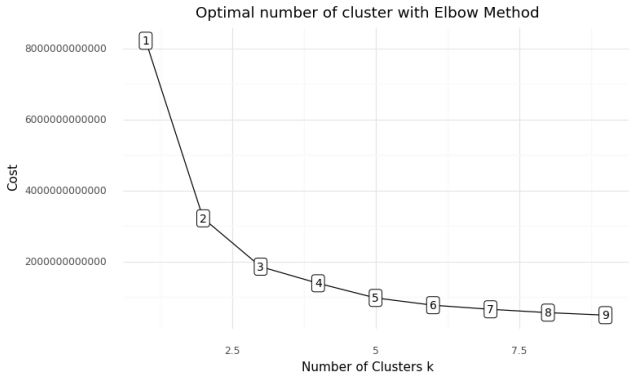
Figure 6

3. Model building

a. K- Prototype Clustering

To better understand buyers' preference and further analyze, I would like to apply clustering method on dataset. I chose **K- Prototype Clustering because our dataset includes both categorical and continuous features.** K-Prototype is a clustering method based on partitioning. Its algorithm is an improvement form of the K-Means and K-Mode clustering algorithm to handle clustering with mixed data types.

The elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use. From the result of elbow method shown below at figure 7, 9 clusters are the best option to minimize cost function.



To interpret the cluster, for the numerical variables, it will be using the average while the categorical using the mode. Table 3 shows us how these 10 clusters look like. Six of them are BMW 3 series because it's the most popular BMW model. But they fall within different prices range and year produced.

	Segment	Total	model	transmission	fuelType	year	price	mileage	tax	mpg	engineSize
0	First	1704	3 Series	Automatic	Diesel	2016.046362	17580.883803	32039.914906	120.240610	62.044836	2.167958
1	Second	457	3 Series	Automatic	Diesel	2013.365427	10227.126915	88006.888403	113.708972	59.898906	2.140481
2	Third	76	3 Series	Automatic	Diesel	2010.973684	6891.250000	129020.105263	162.236842	53.211842	2.234211
3	Forth	838	3 Series	Automatic	Diesel	2014.309069	12669.980907	65340.285203	110.173031	61.634129	2.150477
4	Fifth	1214	3 Series	Automatic	Diesel	2015.321252	14991.121087	46660.160626	112.182867	61.969357	2.152883
5	Sixth	2374	2 Series	Semi-Auto	Diesel	2018.807919	23578.711879	5554.987784	144.220725	54.686521	1.916849
6	Seventh	1840	1 Series	Semi-Auto	Diesel	2016.860870	19110.497283	19323.522283	132.991848	59.805326	2.111957
7	Eighth	592	X5	Semi-Auto	Diesel	2019.106419	55410.423986	4839.454392	146.798986	37.860811	3.043243
8	Ninth	1686	3 Series	Semi-Auto	Diesel	2019.138197	33911.453144	4302.460261	147.206406	48.478233	2.298102

b. Classification Model

Now, let us step into classification part and figure out how to classify a new request into existing cluster. With the dataset and the new data prepared and ready to go, we can begin modeling with our classifiers. In this project, I only consider **Dummy Classifier** (which will function as our baseline model), **KNN Classifier and Support Vector Machine.**

To pick up the best classifier, we use **Macro Average-F1 Score as evaluation metric**. We are using the Macro Average because of the **class imbalance** that is inherent to our dataset and the macro average is sensitive to that imbalance compared the micro average. The clustering algorithm does not guarantee that each cluster contains the same amount of cars. The F1 Score is used because it strikes a good balance between Precision and Recall scores.

After looping through the models and printing out the scores for each model, we are left with the following scores. **The best model with a score of around 77% is the k-nearest neighbors**. We will then be using the KNN classifier to classify our new car request.

```
Dummy (Macro Avg - F1 Score):
0.11580187855307732

KNN (Macro Avg - F1 Score):
0.7737317655303583

SVM (Macro Avg - F1 Score):
0.62377238488818
KNN Score: 0.7737317655303583
```

By fitting the KNN classifier to the entire dataset, then using it to predict the cluster for our new input, we are able to find that the **predicted cluster for our new profile is 8th Cluster**. From there we are able to narrow down the entire dataset to only include those with 8th Cluster.

Once we have done so, we can find the correlations among the cars. After we have the correlations, we can narrow down the data to our new request and **sort by the correlation score**. This will finally give us the top ten cars similar to our new piece of data.

	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	Total	Segment
5337	X4	2019	46450	Semi-Auto	9374	Diesel	145	37.2	3.0	7	Eighth
7277	X5	2019	46875	Semi-Auto	9438	Diesel	145	37.7	3.0	7	Eighth
4424	X5	2019	47440	Semi-Auto	9415	Diesel	145	37.7	3.0	7	Eighth
6644	X5	2019	45880	Semi-Auto	9000	Diesel	145	37.7	3.0	7	Eighth
6699	X5	2019	47213	Semi-Auto	9278	Diesel	145	37.7	3.0	7	Eighth
3941	M4	2019	44980	Semi-Auto	9244	Petrol	145	32.5	3.0	7	Eighth
454	X5	2019	46600	Semi-Auto	9584	Diesel	145	37.7	3.0	7	Eighth
4232	X5	2019	48980	Semi-Auto	9591	Diesel	145	37.7	3.0	7	Eighth
2771	X5	2020	48975	Semi-Auto	9998	Diesel	145	37.7	3.0	7	Eighth
6923	X5	2019	44850	Semi-Auto	9341	Diesel	145	37.7	3.0	7	Eighth

Top 10 recommendations

Table 3

4. Conclusion

Now, we successfully implemented both classification and clustering methods in BMW used car sales data and build our recommender system. I use K- Prototype instead of K-means+ one-hot encoding for following two reasons.

- If we use K-means + one hot encoding it will increase the size of the dataset extensively if the categorical attributes have a large number of categories. This will make the K-means computationally costly.
- Also, the cluster means will make no sense since the 0 and 1 are not the real values of the data. K-modes on the other hand produces cluster modes which are the real data and hence make the clusters interpretable.

The second session of this project is to classify new request into one cluster and find the most relevant cars in that cluster as recommendations. I considered this way instead of clustering directly because this way would give us more flexibility. It's easier to adapt our recommendations to customers' needs and diversify users options rather than limiting our choices in one cluster. I used F1 score as major metric because our dataset is not balance and it took into consideration of misclassified data when evaluating.

The next step of this project is to try more classification methods and compare its performance. Apparently, our champion model with 77% F1 score could be better. In the meantime, considering more variables is the other thing to try for better model performance. 9 features put a limitation on training model. Also, I am still working on displaying this recommender system as a APP. I believe this would let us better understand how this system works. This would be super helpful if I am going to present to business folks.