Recommender system for optimal motorcycles in consumption

Mehrdad Mohammadian
Summer 2023

Our problem

We are a motorcycle-making company and we have a website to sell our motorcycles. Nowadays, we want to recommend the most optimal motorcycles in terms of their fuel consumption. The reason behind this is that the government made a new set of rules to prevent a catastrophic disaster called global warming! They forced us to show products to our customers based on fuel consumption.

Our solution

First, we did a clustering on data of our motorcycle to group them into 3 categories. These categories represent how much these motorcycles are optimal in terms of fuel consumption. In the second stage, we select those motorcycles that are even more optimal based on other technical features that are related to fuel consumption.

Selected features from the main dataset

Overall, we have 28 features. Most of the numerical values have a high correlation with each other. Many of the columns have a high percentage of null values. Additionally, some of them were irrelevant to fuel consumption which is our main concern. So, based on our problem I ended up with the below features:

- Categorical: *Engine cylinder, Engine stroke, Gearbox, Transmission type, Fuel system, Cooling system*
- Numerical: *Fuel capacity (lts), Dry weight (kg)*

Cluster labels

Based on the k-means result, I changed the name of clusters as follows:

```
kmeans_final.groupby('cluster')['Fuel capacity (lts)'].mean()
```

Excellent Fuel Efficiency 9.099857
Good Fuel Efficiency 14.891920
Moderate Fuel Efficiency 19.372158

As you can see from the clustering information, the more efficient the motorcycle, the less fuel capacity it needs.

```
kmeans_final.groupby('cluster')['Dry weight (kg)'].mean()
```

Excellent Fuel Efficiency 103.729770
Good Fuel Efficiency 179.142056
Moderate Fuel Efficiency 331.993913

Similarly, the more efficient the motorcycle, the less its dry weight.

Related features to fuel consumption

These features have a great impact on the fuel consumption of motorcycles.

Engine Stroke:
Engine stroke can have an influence on fuel consumption in motorcycles. Generally, motorcycles with longer-stroke engines tend to offer better fuel efficiency at lower RPMs due to improved torque characteristics.

Engine cylinder:
The design and configuration of an engine's cylinders can indeed impact the optimal fuel consumption of motorcycles and other vehicles. The number of cylinders, their arrangement, and the overall engine displacement can all play a role in determining fuel efficiency.

Fuel system:
Fuel system of a motorcycle can significantly impact fuel consumption. Different fuel systems have varying degrees of control over fuel delivery, combustion efficiency, and air-fuel mixture.

Cooling system:
The cooling system of a motorcycle's engine can have an impact on its optimal fuel consumption. The cooling system plays a crucial role in maintaining the engine's temperature within an efficient operating range. An improperly designed or inefficient cooling system can affect the overall efficiency of the engine, which in turn can influence fuel consumption.

Gearbox:
Gearbox configuration in a motorcycle can indeed impact fuel consumption. The choice of gearbox influences the range of gear ratios available and how effectively the engine's power is utilized.

Transmission type:
Transmission type in a motorcycle can influence fuel consumption. Different transmission types affect how power is delivered from the engine to the wheels, which can impact fuel efficiency.

By looking at the below images we can see that the “Touring” category is in the cluster of Moderate (least efficient motorcycles). “Other” to “Racing” are in the Good cluster and “Offroad” is in the Excellent.

Category	Fuel capacity (lts)	Dry weight (kg)
Touring	19.955756	322.447415
Other	17.082210	214.259259
Sport	15.261429	155.808480
Allround	14.442368	161.057573
Standard	13.104331	178.901384
Racing	11.975992	168.431282
Offroad	10.005664	114.883726

cluster	Fuel capacity (lts)	Dry weight (kg)
Moderate Fuel Efficiency	19.372158	331.993913
Good Fuel Efficiency	14.891920	179.142056
Excellent Fuel Efficiency	9.099857	103.729770

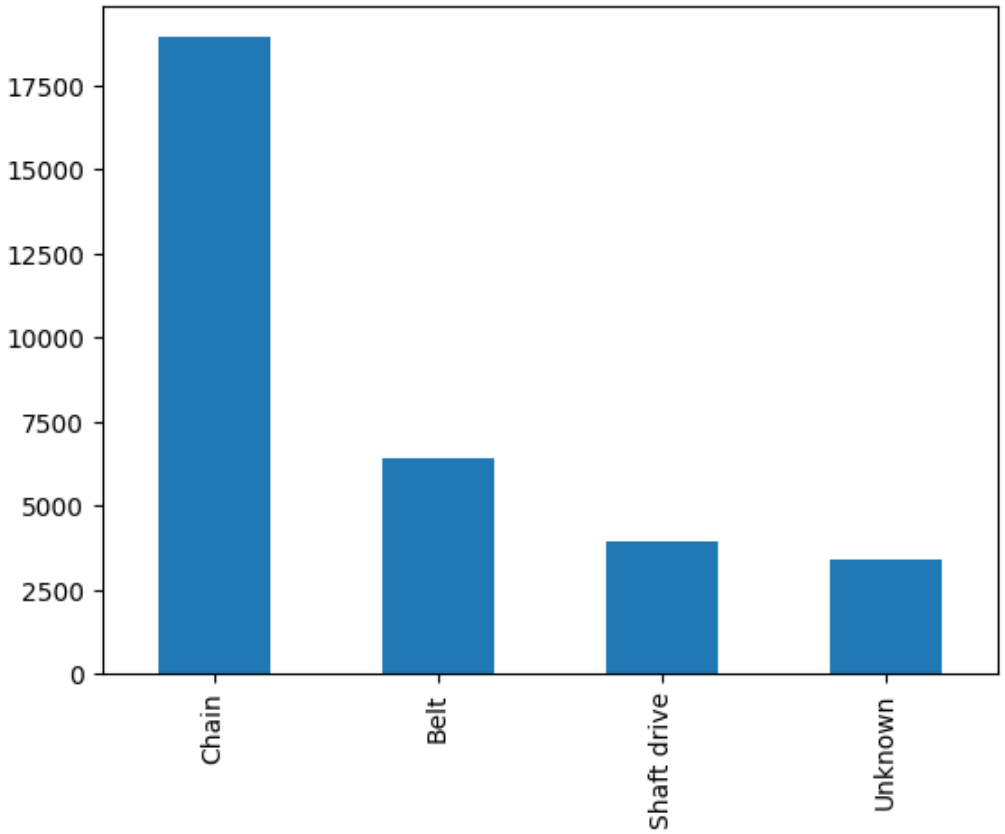
Now, we can see that in the categories of “Other” to “Racing” most of the motors are in the Good cluster. Similarly, the “Offroad” category has the most motorcycles in the cluster of Excellent.

Category cluster	Allround	Offroad	Other	Racing	Sport	Standard	Touring
Excellent Fuel Efficiency	523	4492	8	1069	1247	3474	15
Good Fuel Efficiency	1639	463	756	1650	4851	8933	128
Moderate Fuel Efficiency	22	7	40	457	109	1403	1389

Handling null values

Categorical:
In our case, we can replace the null values of our categorical features with the value of “unknown”. If we wanted to simply drop those rows we would lose a lot of beneficial information. Moreover, even by replacing null values in this way, we did not make a great impact on the frequency ranks of the categorical variables that can be seen in their bar charts.

For example in **`Transmission type`**:



Numerical:
Because **`Dry weight (kg)`** is an important feature in our problem, we have to handle its null values. And because of having categories of motorcycles, it makes sense to replace null values with the mean of their corresponding category. The shape of the distribution in our numerical values did not change a lot by these imputations. An example of **`Fuel capacity (lts)`** is represented in the notebook.

Merging categorical values

By exploring our categorical values we can find out that in many of them, there are some rows that are actually the same but have different values. So, I decided to merge them into more general categories based on their common value. For example, in **`Fuel system`** we have a lot of rows that are started with “Injection”. They represent same information so we can merge them. Similarly, in the feature of “category” we can group different categories into a more general category. It can also help us to reduce the problem of high cardinality. To do this, I used both Google Search and ChatGPT to find out which categories can merge together (are related to each other).

Categorical encoding

I tried both one-hot and frequency encoding methods to handle dummy values. Finally, I chose one-hot encoding for two reasons:

1. The number of my categorical features is relatively small, so we do not suffer from huge numbers of columns after one-hot encoding.
2. I tried both of them and I found out that in this case, I just get better results with one-hot encoding!

Recommendation system

After clustering the data, I decided to use a particular library called “Annoy” that provides an efficient algorithm to build a recommender system. It actually implements a concept called “Approximate Nearest Neighbor” which is more efficient than the normal nearest neighbor. It is useful for situations like mine where resources like memory and CPU are low.

The argument of **`n_trees`** is for choosing the number of trees in the forest. More trees give higher precision when querying. I have found that with n_tree of 1000, we can get relatively good results. We finally printed the top 10 recommendations for the input.

https://github.com/spotify/annoy
https://en.wikipedia.org/wiki/Nearest_neighbor_search#Approximate_nearest_neighbor

Silhouette Score: 0.64
And also I have got this number for the Silhouette.

A recommendation example

Target Motorcycle:

Category	Standard
Engine cylinder	In-line four
Engine stroke	four-stroke
Gearbox	6-speed
Fuel capacity (lts)	13.442412
Fuel system	Turbo
Cooling system	Liquid
Transmission type	Unknown
Dry weight (kg)	300.0
cluster	Moderate Fuel Efficiency
Motorcycle_ID	2

Recommended Motorcycles:

Category	Sport
Engine cylinder	In-line four
Engine stroke	four-stroke
Gearbox	6-speed
Fuel capacity (lts)	13.442412
Fuel system	Turbo
Cooling system	Liquid
Transmission type	Unknown
Dry weight (kg)	360.0
cluster	Moderate Fuel Efficiency
Motorcycle_ID	1

Distance Score: 0.0

As you can see, they are quite similar! Both are in the same cluster.
Distance Score of 0.0 means that this sample is too close to our target sample that we want to get a recommendation for it.

Next recommendation with a bigger distance score:

Category	Touring
Engine cylinder	In-line four
Engine stroke	four-stroke
Gearbox	6-speed
Fuel capacity (lts)	13.442412
Fuel system	Injection
Cooling system	Liquid
Transmission type	Unknown
Dry weight (kg)	322.447415
cluster	Moderate Fuel Efficiency
Motorcycle_ID	14571

Distance Score: 0.006068084854632616

PCA and other kinds of stuff!

Well, this task has a problem called density! I mean it does not matter to choose which set of features to do clustering you always face a big dense cluster …. However, I tried my best to make more sensible and logical results from it!

In regards to PCA, it does not make any difference in silhouette by applying it before the k-mean or not. By the way, I decided to apply PCA because, in my previous run when I was working with another set of columns, I saw that with PCA I got relatively better results.

I did not use any scaler for my numerical values. Because I believe that each of them has their unique meaning. So I did not want to make them similar to each other, it may help us to avoid any misunderstanding for k-means.

For feature selection, I first used a library called pandas-profiling. It helped me very well to gain a general insight into my data. Next, I checked the null values to see which columns I should drop because we do not want to handle that many null values for all columns! After that, I set my problem and selected my columns in response to it. Again I used Google Search and ChatGPT to find out which features are useful to solve this problem.

I actually tried both the Brich and K-means algorithms, but in this problem that I am trying to solve their results are too similar. They both have a silhouette score of 0.64 and also mean of clusters in numerical variables is too similar.

Thanks for your time and consideration! :)