**汽车车型及其竞品分析**

**摘要：** 基于阿里天池提供的真实车型数据集，针对竞品分析和车型画像展开了分析。采用多种方法对数据进行处理和分析，以揭示车型之间的关联性和特征。特别地，本文以一款特定车型为例，通过多种聚类分析的方法对比分析，成功地找到了与之具有相似特征的竞品车型。这一研究成果为产品的定位和竞品分析提供了有力的数据决策支持，具备显著的实际应用价值。

1. **数据获取与介绍**
2. **数据获取**

本题数据来自阿里天池提供的真实车型数据集car\_price.csv，数据包括了205款车的26个字段。

1. **数据介绍**

下面是数据的每一个字段的详细介绍：

表 1 数据集字段详细介绍

|  |  |  |
| --- | --- | --- |
| 1 | ****Car\_ID**** | **Unique id of each observation (Interger)** |
| 2 | **Symboling** | Its assigned insurance risk rating, A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe.(Categorical) |
| 3 | **carCompany** | Name of car company (Categorical) |
| 4 | **fueltype** | Car fuel type i.e gas or diesel (Categorical) |
| 5 | **aspiration** | Aspiration used in a car (Categorical) |
| 6 | **doornumber** | Number of doors in a car (Categorical) |
| 7 | **carbody** | body of car (Categorical) |
| 8 | **drivewheel** | type of drive wheel (Categorical) |
| 9 | **enginelocation** | Location of car engine (Categorical) |
| 10 | **wheelbase** | Weelbase of car (Numeric) |
| 11 | **carlength** | Length of car (Numeric) |
| 12 | **carwidth** | Width of car (Numeric) |
| 13 | **carheight** | height of car (Numeric) |
| 14 | **curbweight** | The weight of a car without occupants or baggage. (Numeric) |
| 15 | **enginetype** | Type of engine. (Categorical) |
| 16 | **cylindernumber** | cylinder placed in the car (Categorical) |
| 17 | **enginesize** | Size of car (Numeric) |
| 18 | **fuelsystem** | Fuel system of car (Categorical) |
| 19 | **boreratio** | Boreratio of car (Numeric) |
| 20 | **stroke** | Stroke or volume inside the engine (Numeric) |
| 21 | **compressionratio** | compression ratio of car (Numeric) |
| 22 | **horsepower** | Horsepower (Numeric) |
| 23 | **peakrpm** | car peak rpm (Numeric) |
| 24 | **citympg** | Mileage in city (Numeric) |
| 25 | **highwaympg** | Mileage on highway (Numeric) |
| 26 | **price(Dependent variable)** | Price of car (Numeric) |

1. **研究目标**

我们的目标是通过多种方法对真实的车型数据进行处理和分析，以揭示车型之间的关联性和特征，并通过聚类分析找到指定车型的竞品车型。通过深入挖掘车型数据，我们的研究旨在为汽车行业的市场定位和竞争策略提供新的视角和方法，以推动行业的发展。

1. 数据探索
2. 数据特征分析

通过python的duplicated().sum()与isnull().sum()分别对重复值和空值进行检查，发现并不存在重复值和空值。

利用python的info()函数可以把数据特征区分为3大类：

第一类：汽车ID类属性：

表 2 汽车ID属性

|  |  |
| --- | --- |
| ID | 说明 |
| 1 | Car\_ID 车号 |
| 3 | CarName 车名 |

第二类：类别型变量（10个）

表 3 类别型变量

|  |  |
| --- | --- |
| ID | 说明 |
| 2 | Symboling 保险风险评级 |
| 4 | fueltype 燃料类型 |
| 5 | aspiration 发动机吸气形式 |
| 6 | doornumber 车门数 |
| 7 | carbody 车身型式 |
| 8 | drivewheel 驱动轮 |
| 9 | enginelocation 发动机位置 |
| 15 | enginetype 发动机型号 |
| 16 | cylindernumber 气缸数 |
| 18 | fuelsystem 燃油系统 |

第三类：连续数值型变量（14个）

表 4 连续数值型变量

|  |  |
| --- | --- |
| ID | 说明 |
| 10 | wheelbase 轴距 |
| 11 | carlength 车长 |
| 12 | carwidth 车宽 |
| 13 | carheight 车高 |
| 14 | curbweight 整备质量（汽车净重） |
| 17 | enginesize 发动机尺寸 |
| 19 | boreratio 气缸横截面面积与冲程比 |
| 20 | stroke 发动机冲程 |
| 21 | compressionratio 压缩比 |
| 22 | horsepower 马力 |
| 23 | peakrpm 最大功率转速 |
| 24 | citympg 城市里程（每加仑英里数） |
| 25 | highwaympg 高速公路里程（每加仑英里数） |
| 26 | price(Dependent variable) 价格（因变量） |

1. 检查变量特征取值情况
   1. 查看类别属性特征分类取值情况

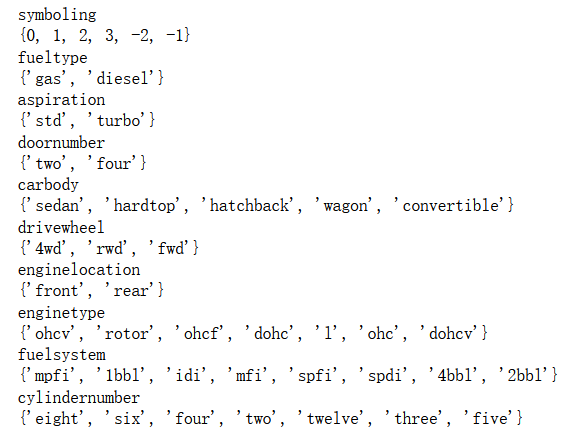


图 1类别属性特征分类取值情况

由上面可检查类别型特征数据是否有拼写错误，还可知道特征的具体分类情况；

分类取值具有大小意义的，如：

①保险风险评级Symboling的取值范围为：0、 1、2、3、-2、-1，虽是分类特征但其取值是有大小意义的；

②气缸数cylindernumber取值：{‘three’, ‘six’, ‘eight’, ‘five’, ‘four’, ‘twelve’, ‘two’}，这7个取值也是有大小意义的，在同等缸径下，缸数越多，排量越大，功率越高；在同等排量下，缸数越多，缸径越小，转速可以提高，从而获得较大的提升功率；

其他分类取值没有大小意义的，如：

车门数doornumber分’two’、 'four’两类，因车门数是跟车外形设计有关，如公务用途的轿车为四门，而运动用途跑车为两门，完全是不同类型的车型，其取值没有大小意义，只是分类；

fueltype 燃料类型分’gas’和’diesel’两类，等等只是类别上属性的分类。

可看到有很多命名是不规则的，需要修正。如：‘toyouta’, ‘maxda’, ‘porcshce’,‘Nissan’, ‘vw’,‘vokswagen’，我们根据实际情况对其进行修正处理。

* 1. 检查数值型数据

在剔除汽车ID类属性使用describe()函数对数据进行描述性统计：

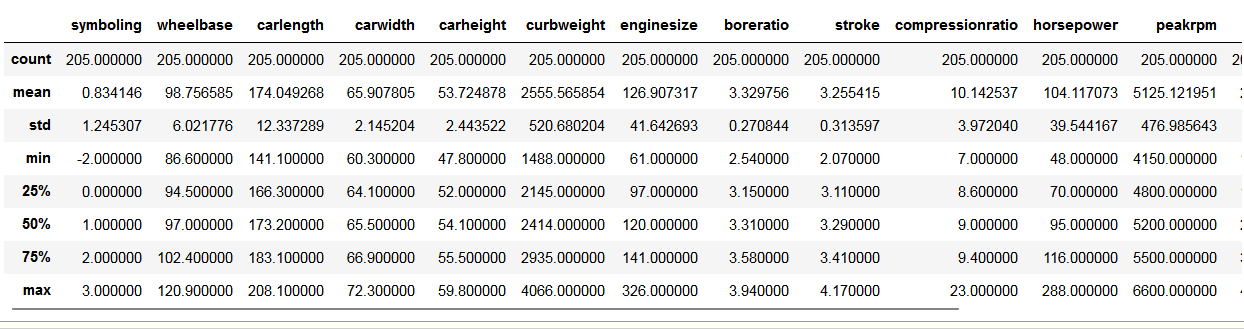


图 2 数据描述性统计

初步查看发现没有明显异常值，为了进一步分析数据，我们采用绘制箱线图的方式查看异常值：

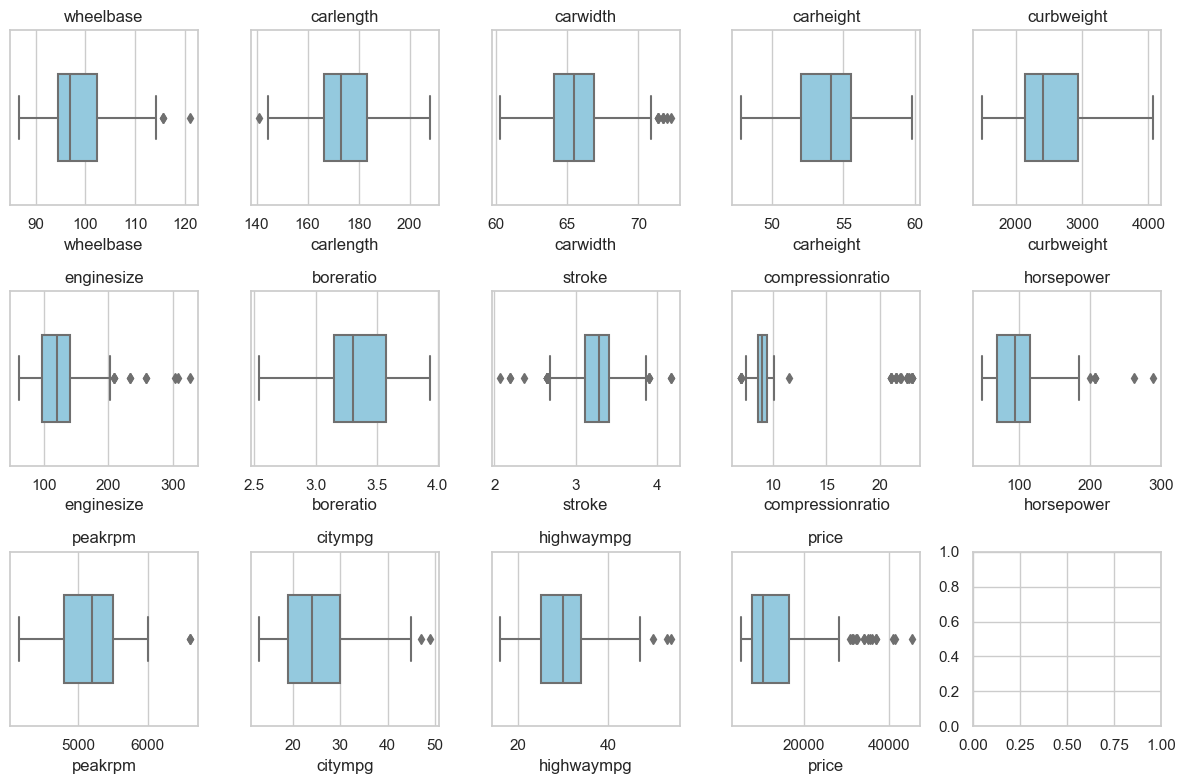


图 3 数值型数据箱线图

对各特征的箱线图进行观察可知，部分特征存在离群点，但不存在特别明显的离群点，不需要进行异常处理。

* 1. 检查特征数据之间的逻辑关系

进行特征工程，分析各个特征之间是否存在逻辑关系，是否可以进行数据特征融合或拆分等操作。

* + 1. 由Carname特征拆分品牌信息

由CarName数据组成信息，第一个英文为其车型的品牌，使用split()函数将其拆分出来。在处理数据的时候，我们发现CarName部分命名不规则，很多车名与现实情况的车名有一定差异，分析后考虑纠正这些拼写错误。

* + 1. 根据车长划分车型大小

按德国标准，车型大小可按照车长，轴距划分为6类：

表 5 汽车分类等级

|  |  |  |
| --- | --- | --- |
| **等级** | **车长** | **轴距** |
| 微型车A00 | 车长小于3.7M | 轴距在2米至2.2米 |
| 小型车（A0） | 车长小于4.3M | 轴距为2.3米至2.45米 |
| 紧凑型车（A） | 车长小于4.6M | 轴距在2.45米至2.6米 |
| 中型车（B） | 车长小于4.9M | 轴距约在2.6米至2.75米 |
| 中大型车（C） | 车长小于5.1M | 轴距约在2.7米至2.8米 |
| 豪华车（D） | 车长大于5.1M | 轴距一般均大于2.8米 |

而要注意，数据集中车长宽高和轴距单位均为英寸，需要进行单位的转换：1英寸=0.0254米。

按车身长度分类界限：微型车： A00 <145.67 ；小型车： A0 <169.29 ；紧凑型车：A <181.10 ；中型车： B <192.91 ；中大型车：C <200.79 ；大型车： D >200.79

创建新的特征CarSize，为Categories (6, object): [‘A00’ < ‘A0’ < ‘A’ < ‘B’ < ‘C’ < ‘D’]，其取值有大小的意义，当有车型大小分类后，选择特征聚类时，车身的长和宽可剔除，而在同类车型中车高和轴距则可当为车身空间舒适性度量来分析。

* + 1. 数值型数据相关性分析

查看数值型特征的与价格的相关系数：

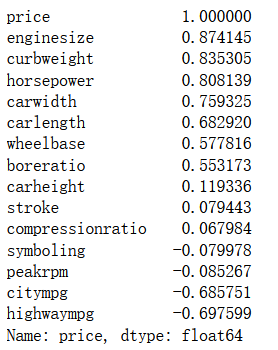


图 4 数值型特征的与价格的相关系数

绘制相关性热力图：

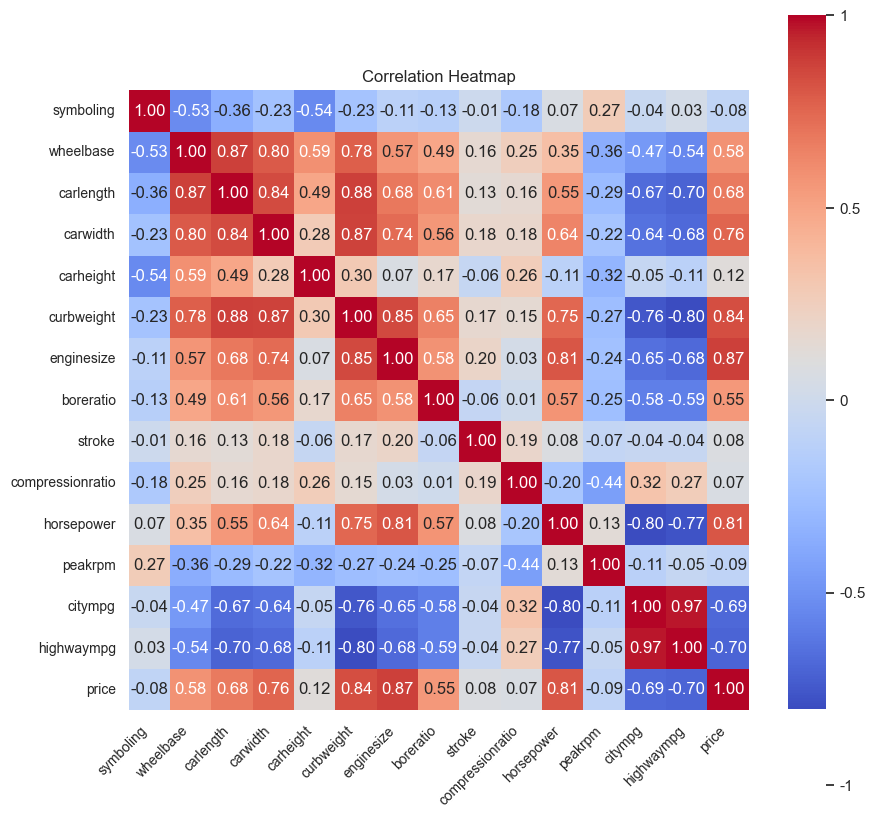


图 5 相关性热力图

相关系数分类： 0.8-1.0 极强相关；0.6-0.8 强相关；0.4-0.6 中等程度相关；0.2-0.4 弱相关；0.0-0.2 极弱相关或无相关

由上面热力图可看出： 车长、宽、轴距三者都极强相关，整备质量和车长、宽、发动机尺寸极强相关，价格与车整备质量、发动机尺寸、马力具有极强相关性，等等。部分数据之间存在高度相似，数据存在冗余，需要进行数据处理操作。

1. 数据预处理
2. 选择合适的特征

在前面我们已经创建CarSize特征，车身的长和宽可剔除

1. 对类别型变量进行数值映射和one-hot编码

主要策略如下：

1. 将取值具有大小意义的类别型变量数据转变为数值型映射
2. 使用LabelEncoder对不具实体数值数据编码
3. 对于类别离散型特征，取值间没有大小意义的，可采用one-hot编码

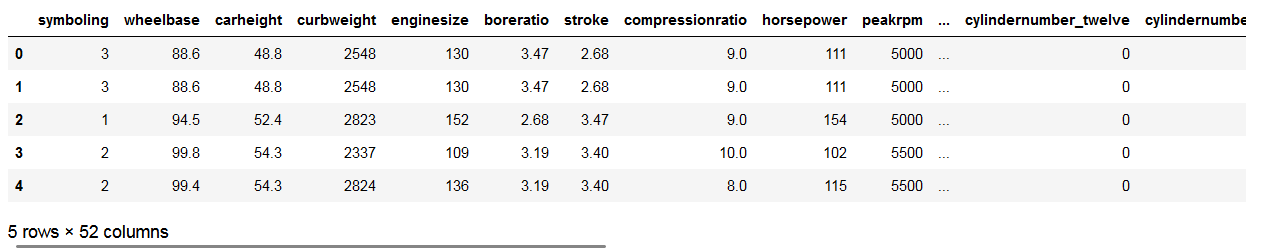


图 6 数据预处理后的特征

1. 对数值型变量归一化处理

特征归一化（Feature Normalization）是一种常见的数据预处理技术，用于将不同特征之间的取值范围进行统一，以消除特征之间的量纲差异。特征归一化有助于提高模型的性能和收敛速度，并且可以使优化算法更稳定。

在该数据集中数据变量之间量级差异比较大，需要进行数据归一化。我们采用的方法是最小-最大缩放（Min-Max Scaling）法。

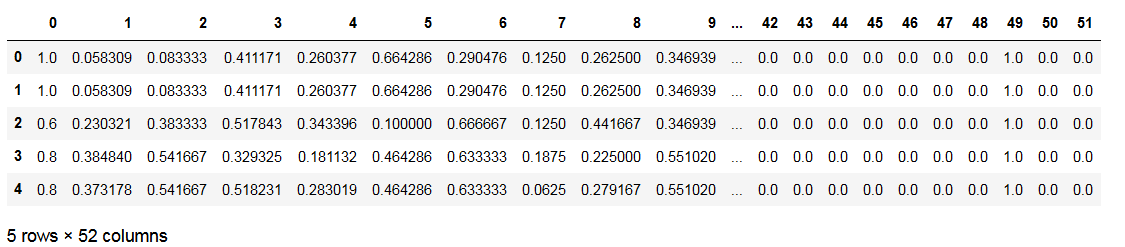


图 7 归一化后的数据

1. 数据降维

数据降维（Dimensionality reduction）是指通过减少数据特征的数量，从而减少数据集的维度，同时保留或捕捉数据集中的关键信息。数据降维的目的是减少数据的复杂性、降低计算成本、提高模型的效率，并帮助可视化和理解数据。

我们采用的方式是PCA（主成分分析聚类方法），降维后的特征是原始特征的线性组合。

首先我们设定降维后的数据保持90%的信息，查看在该情况下保留的特征数量与每个主要成分的解释方差占比：

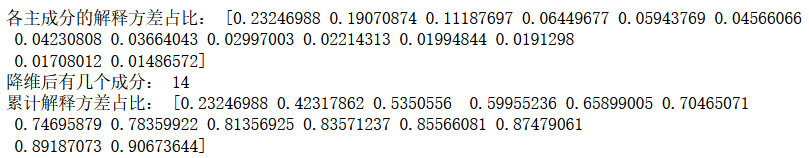


图 8 降维后的数据解释方差

可视化处理：

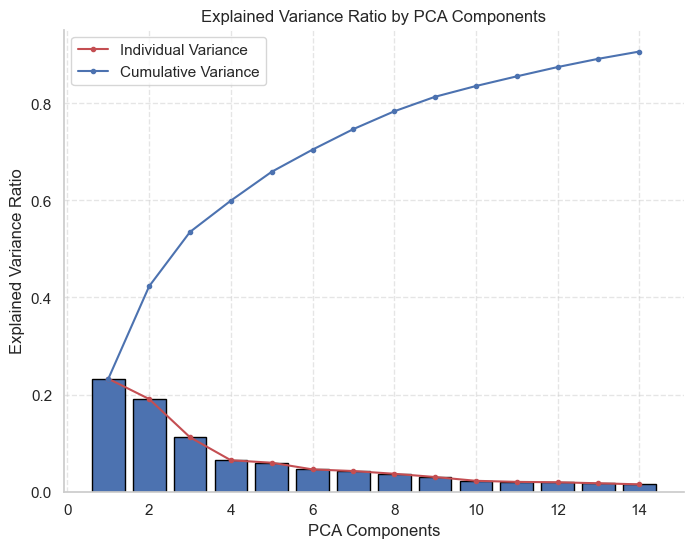


图 9 降维数据可视化处理

蓝色折线为累计方差占比，红色折线为每个成分方差占比。

1. 聚类模型的建立

对于聚类，常用的算法包括K均值聚类（K-means clustering）、层次聚类（Hierarchical clustering）和DBSCAN等。选择适当的聚类算法需要考虑数据的分布、样本量和聚类的目标等因素。通常，K均值聚类适用于数据分布明显的情况，层次聚类适用于数据具有层次结构的情况，而DBSCAN适用于发现任意形状的聚类簇。

1. K均值聚类（K-means clustering）

优点：

1. 简单而高效：K均值聚类算法简单易懂，计算效率高，尤其适用于大规模数据集。
2. 可扩展性强：K均值聚类可以适应不同的数据集和问题，并且可以灵活调整簇的数量。
3. 易解释性：K均值聚类的结果相对容易解释和理解，簇中心代表了各个簇的特征。
4. 适用性广泛：K均值聚类可以用于各种数据类型和特征空间，适用于连续型和离散型特征。

缺点：

1. 初始簇中心敏感性：K均值聚类对初始簇中心的选择非常敏感，不同的初始值可能会导致不同的聚类结果。
2. 受离群点影响：离群点的存在可能会对K均值聚类产生较大的影响，导致簇中心偏移或聚类结果不准确。
3. 需要预先指定簇的数量：K均值聚类在开始之前需要事先指定簇的数量K，但在某些情况下，选择合适的K值并不容易。
4. 对初始分配敏感：K均值聚类的初始样本分配是随机的，可能会导致不同的初始分配结果产生不同的聚类结果。
   1. 利用肘方法确定簇的最佳数量

肘方法是一种常用的确定聚类簇最佳数量的方法。它通过绘制聚类簇数量和对应的聚类性能指标之间的关系图，找到一个“肘点”，即在该点处进一步增加簇的数量不再显著提高聚类性能。通过观察关系图，选择肘点对应的簇数量作为最佳的聚类簇数量。

我们综合考虑SSE（离差平方和）和轮廓系数来选择最佳的簇数量：

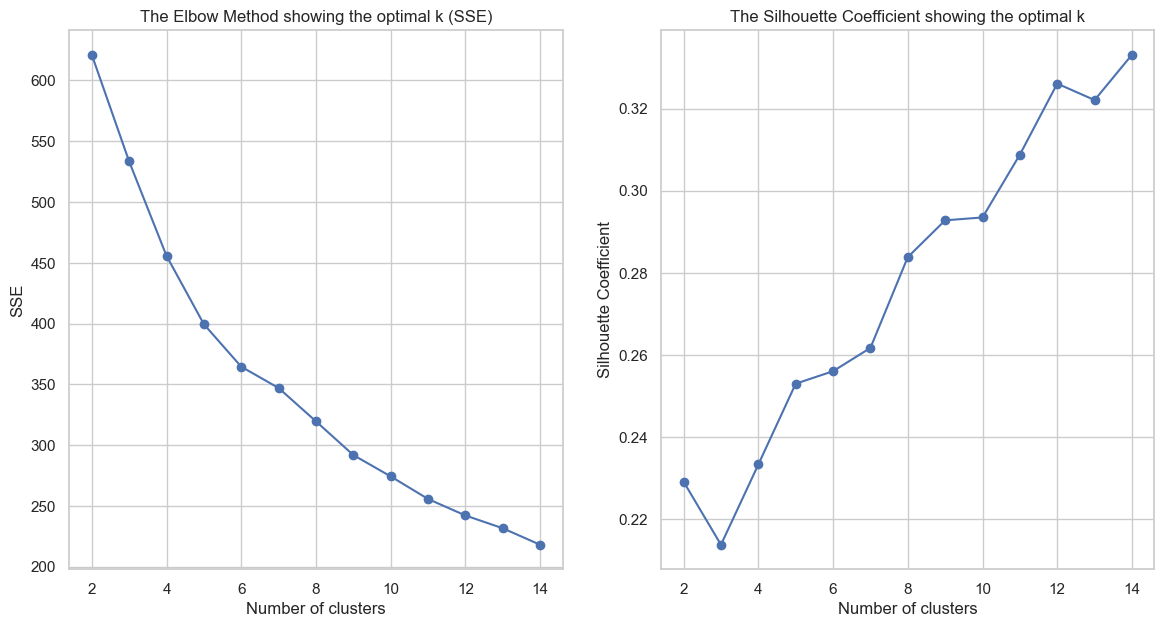


图 10 不同簇数与SSE和轮廓系数的关系变化图

综合两张图，我们发现第一张的图的拐点在9，同时簇数为9时轮廓系数的值也比较大，故最终选定簇数为9。

* 1. 应用K-means聚类模型

使用python sklearn库中的Kmeans模型，设定簇为9，进行聚类，并进行可视化：

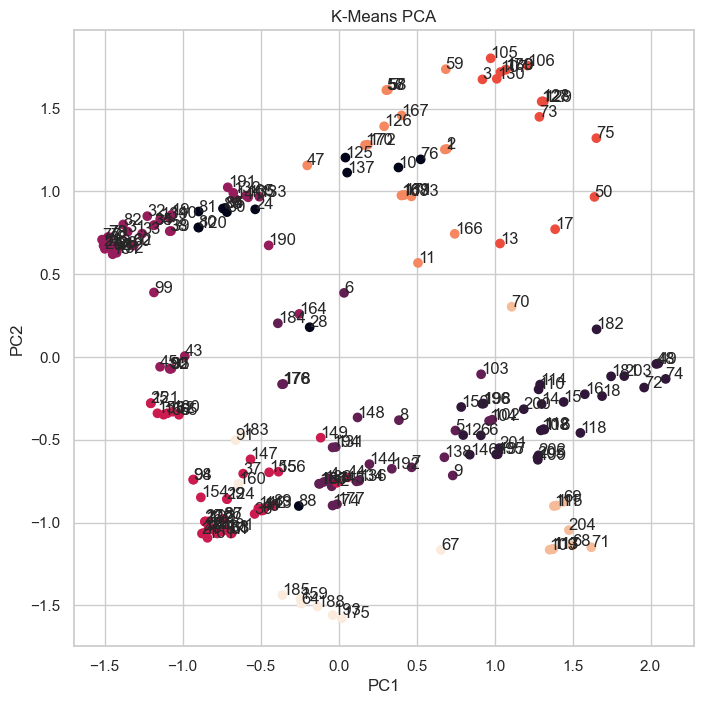


图 11kmeans二维聚类可视化

上面以占比最大的前2个主成分画出的二维散点图，发现部分聚类点距离很远，效果似乎不是很好；考虑到保留的主成分为9个，且前两个主成分累计可解释方差占比仅约为42%，损失信息较多，所以尝试利用前3个主成分画3d效果图再查看。

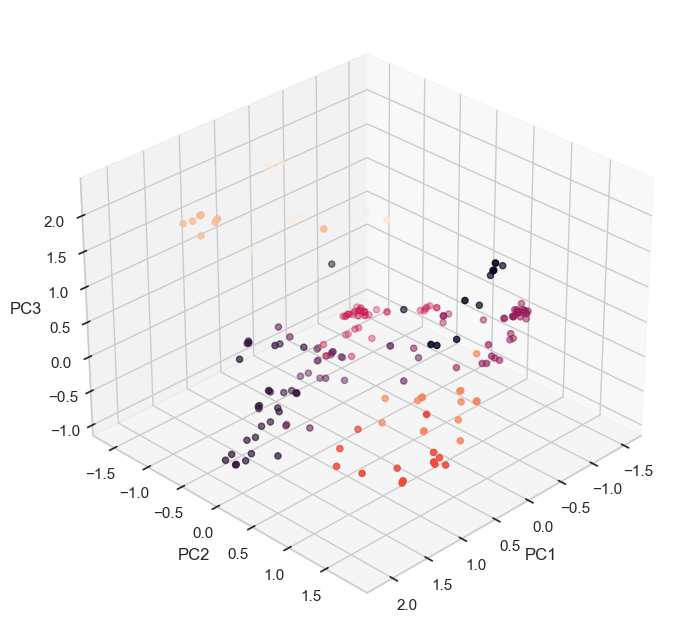


图 12 kmeans三维聚类可视化

从三维散点图来看，效果很不错。

1. 层次聚类（Hierarchical clustering）

层次聚类法是一种聚类分析方法，它通过构建一个多级的嵌套簇树来对数据进行分层次的划分。与K-Means这类划分聚类方法不同，层次聚类不需要事先指定簇的数量，而是生成一个簇的层次结构。层次聚类主要有两种形式：凝聚的层次聚类（Agglomerative Hierarchical Clustering）和分裂的层次聚类（Divisive Hierarchical Clustering）。

优点：

1. 不需要预先指定簇的数量：层次聚类不需要事先指定聚类的数量，它会自动构建一个树状结构，根据数据的相似性将样本逐步合并为簇。
2. 提供层次结构和可视化：层次聚类提供了一个层次结构，可以清晰地显示数据之间的相似性和聚类的关系。这种可视化可以帮助理解数据的组织结构。
3. 灵活性：层次聚类可以适应不同的数据类型和距离度量方法，因此对于各种类型的数据集都具有一定的适应性。
4. 可以处理小样本数据集：相对于其他聚类算法，层次聚类在处理小样本数据集时通常表现较好。

缺点：

1. 计算复杂度高：层次聚类的计算复杂度较高，尤其是对于大规模数据集而言，需要消耗大量的计算资源和时间。
2. 对噪声和离群点敏感：层次聚类对噪声和离群点比较敏感，可能会导致错误的合并或分裂，从而影响聚类结果的准确性。
3. 难以处理大规模数据集：由于计算复杂度高，层次聚类往往难以处理大规模数据集，可能会导致内存和计算资源的限制。
4. 依赖距离度量方法：层次聚类的结果受到所选择的距离度量方法的影响，不同的距离度量方法可能会导致不同的聚类结果。

我们采用的方法是凝聚的层次聚类方法，由于层次聚类的性质，我们不会像在K-Means中那样直接得到WCSS，而是需要手动计算每个簇内样本与其质心之间距离的平方和。

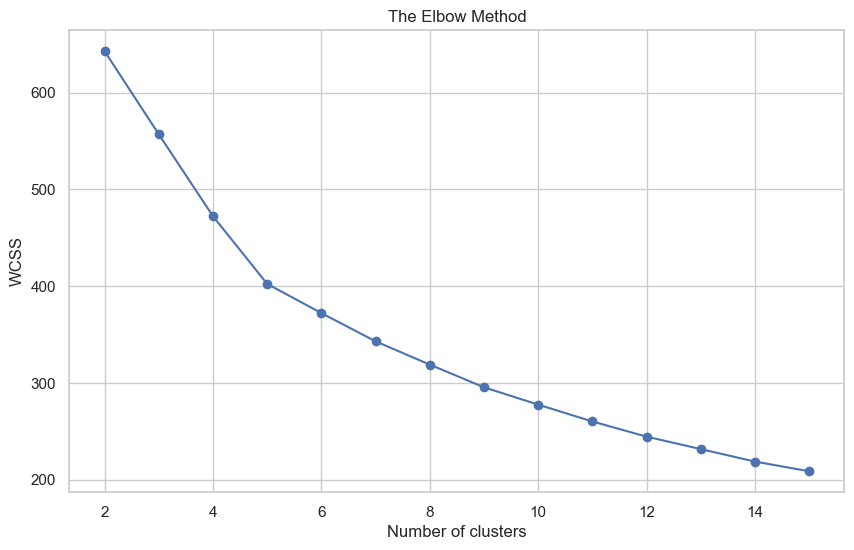


图 13 层次聚类法肘方法图

从图中不难发现，拐点在簇为5时，故我们选择的簇数为5。使用sklearn库中的AgglomerativeClustering库进行层次聚类，可视化结果如下：

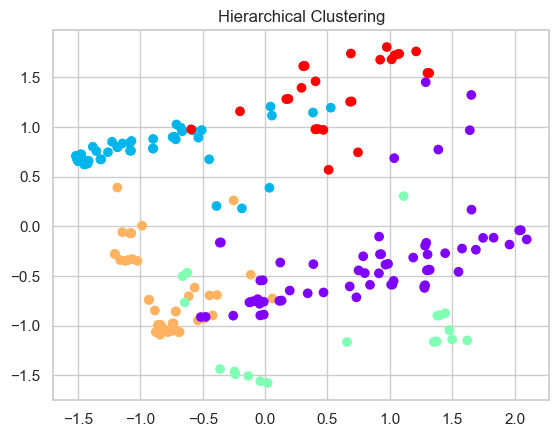


图 14 层次聚类散点图

1. DBSCAN聚类法

DBSCAN（Density-Based Spatial Clustering of Applications with Noise）是一种基于密度的聚类算法，它能够识别任意形状的簇，并且对噪声点具有良好的鲁棒性。这使得DBSCAN在许多实际应用中非常有用，尤其是当簇的形状复杂或者存在异常点时。

优点：

1. 能够发现任意形状的聚类：DBSCAN不受聚类形状的限制，可以有效地发现具有任意形状的聚类，包括凸形、非凸形和嵌套形状。
2. 可以处理噪声和离群点：DBSCAN能够将噪声和离群点标记为噪声簇，并将其排除在聚类结果之外，从而提供了对异常值的鲁棒性。
3. 不需要预先指定簇的数量：DBSCAN不需要事先指定聚类的数量，它可以自动确定簇的数量和大小，更适用于不确定聚类数量的情况。
4. 对参数的鲁棒性较强：DBSCAN只有两个主要参数，即邻域半径和最小样本数，对于大多数数据集，这两个参数的选择相对较为稳定。

缺点：

1. 对密度不均匀数据集敏感：DBSCAN对密度不均匀的数据集的聚类效果可能不理想，对于密度变化较大的数据集，可能会导致错误的聚类结果。
2. 需要调整参数：尽管DBSCAN对参数的鲁棒性较强，但正确设置邻域半径和最小样本数仍然是一个挑战，不同的参数选择可能会导致不同的聚类结果。
3. 对高维数据集的效果较差：DBSCAN在处理高维数据集时，由于维度灾难的影响，效果可能较差。高维空间中的数据点往往分布稀疏，难以满足密度聚类的要求。
4. 对数据集的可伸缩性有限：DBSCAN的计算复杂度较高，特别是在大规模数据集上，可能会受到内存和计算资源的限制。

DBSCAN对eps和min\_samples参数非常敏感。选择不当可能会导致所有点都被归为噪声，或者整个数据集被归为一个簇。我们需要根据数据的密度和分布特征进行多次尝试，以找到合适的参数值。

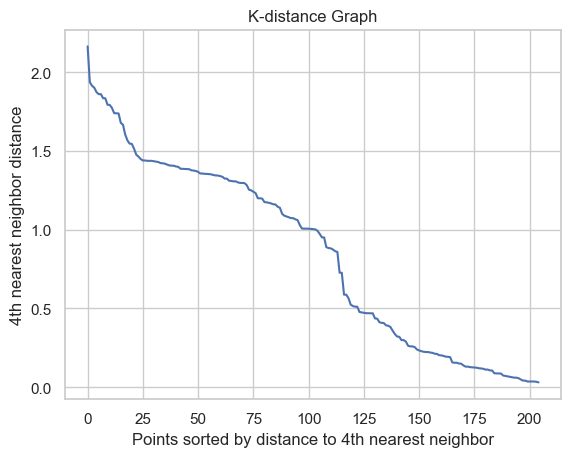


图 15 寻找最优eps参数

最后确定：eps=0.5, min\_samples=5，进行聚类可视化处理：

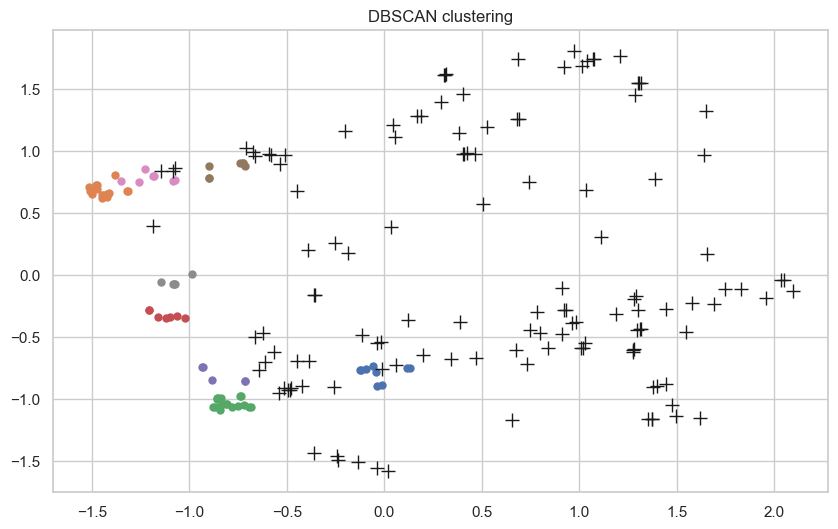


图 16 DBSCAN聚类结果

自动聚类形成的簇为8，并识别出131个噪声点。

1. 模型对比评估

对比层次聚类、K-Means聚类和DBSCAN聚类效果的一个常见方法是使用定量的评估指标。

#### 轮廓系数（Silhouette Coefficient）：

轮廓系数结合了聚类的紧密度和分离度，其值范围为[-1, 1]。值越高，表示聚类效果越好。我们使用sklearn.metrics.silhouette\_score函数计算轮廓系数，并将结果打印输出：

其中K-Means聚类轮廓系数为：**0.2928** ；

层次聚类聚类轮廓系数为：**0.2255;**

DBSCAN聚类轮廓系数为: **0.1587。**

1. **Davies-Bouldin指数：**

Davies-Bouldin指数是一种用于评估聚类结果的指标，用于衡量聚类的紧密度和分离度。它基于簇内的平均相似度和簇间的平均相异度，通过计算簇之间的平均相似度与簇内的平均相异度的比值来评估聚类的质量。Davies-Bouldin指数的取值范围从0到正无穷，值越小表示聚类结果的紧密度和分离度越好，聚类效果越优秀。

使用sklearn.metrics.davies\_bouldin\_score函数计算Davies-Bouldin指数，并将结果打印输出。

其中K-Means聚类轮廓系数为：**1.3075** ；

层次聚类聚类轮廓系数为：**1.4886;**

DBSCAN聚类轮廓系数为: **1.1501。**

综合两个指标来看，kmeans聚类表现最好，故我们选择kmeans聚类法进行下一步分析操作。

1. 聚类结果展示
2. 聚类结果统计

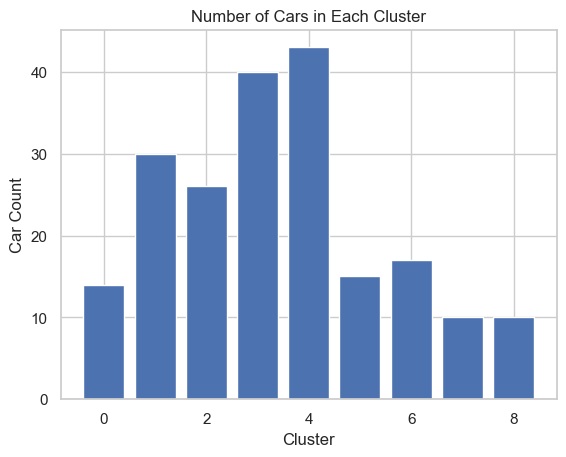


图 17 每个类车型数量

1. 找到任意车型竞品

以宝马热门车型“bmw x1”为例，我们来分析它的竞品：

查看发现“bmw x1”的聚类群为5，找出聚类群5中所有车型：

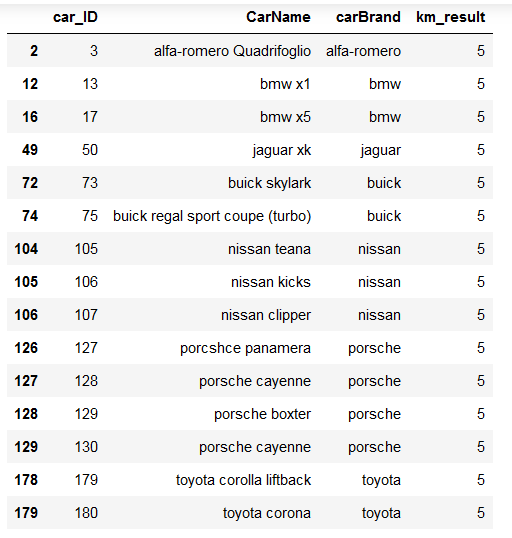


图 18 聚类5中所有汽车

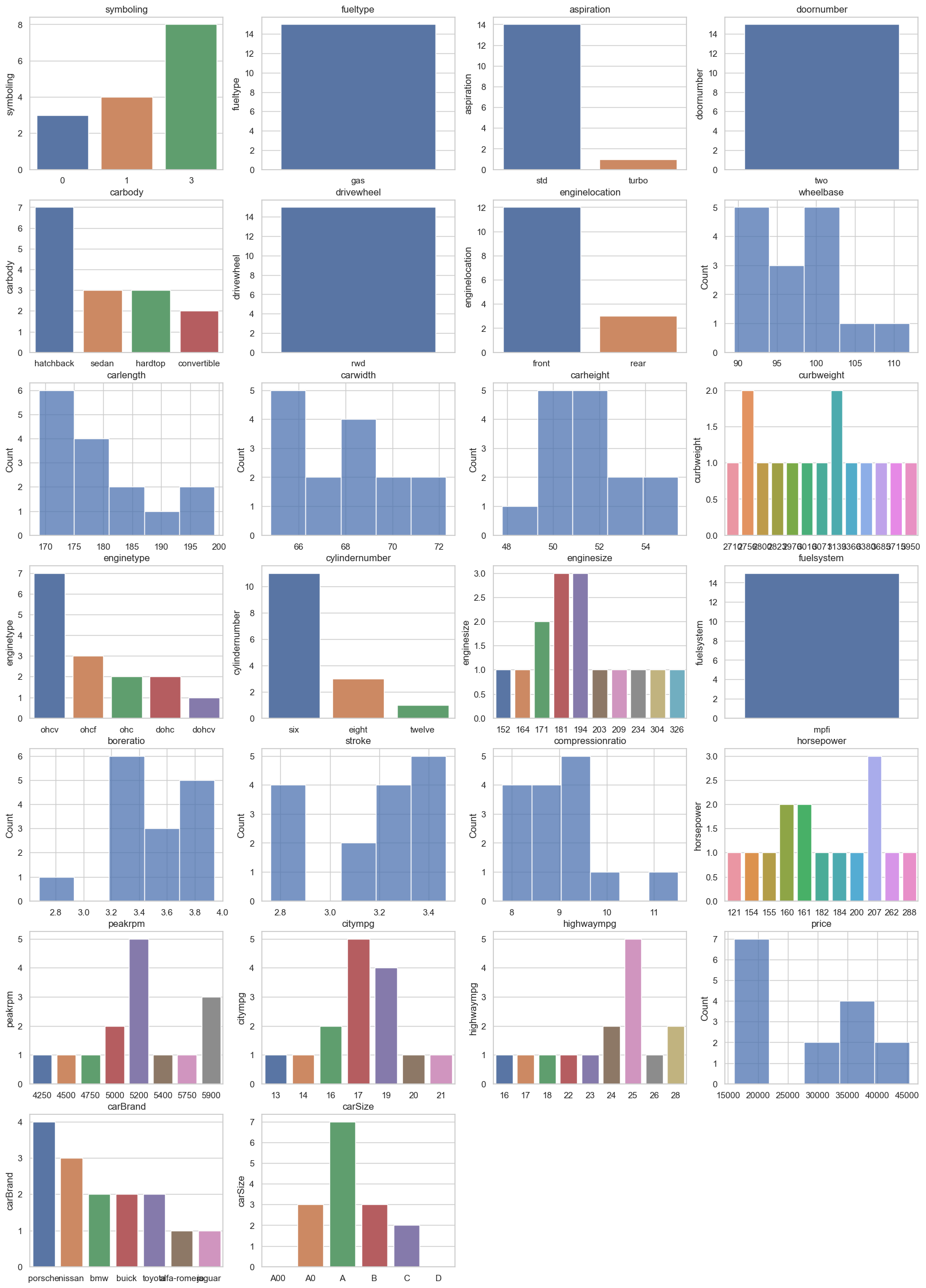


图 19 集群5的车型所有特征分布

由集群5的变量特征分布图可知，类别型变量取值只有一种的有：fueltype : {‘gas’}；doornumber : {‘two’}；fuelsystem : {‘mpfi’}；这些共性的特征在竞品分析时可不考虑。

根据乘用车的汽车用户需求特点，一般首先会考虑车型大小级别；而乘用车的对比也是基于同等级别的车型大小进行.我们只需要对比集群5中carSize为A的车型就行。

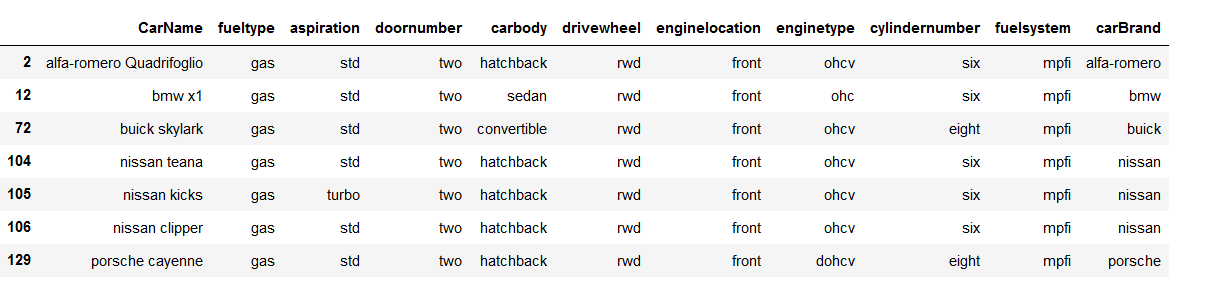


图 20 A级车型的类别型变量的分类情况

综合我们不难发现“bmw x1”的竞品车型为：双开门燃油系统为mpfi的天然气车，并且驱动轮都为rwd，发动机都是前置。