目录

1 数据预处理

在我们拿到数据后，我们先人为地浏览了数据的大致情况，并在此之后进行了预处理。

1.1 数据预览

我们先统计了各个数据的缺失值，如图1.1和图1.2所示。

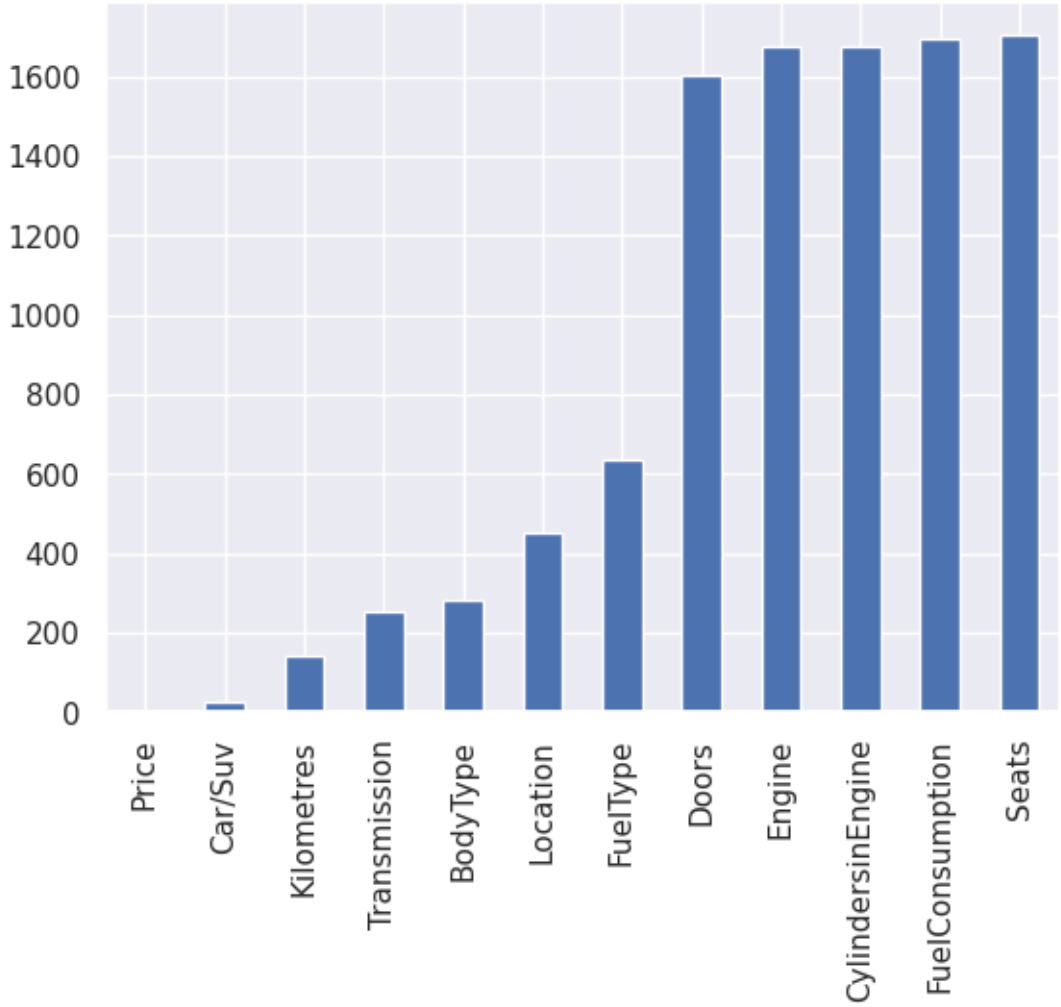


图 1.1: 缺失值数量统计

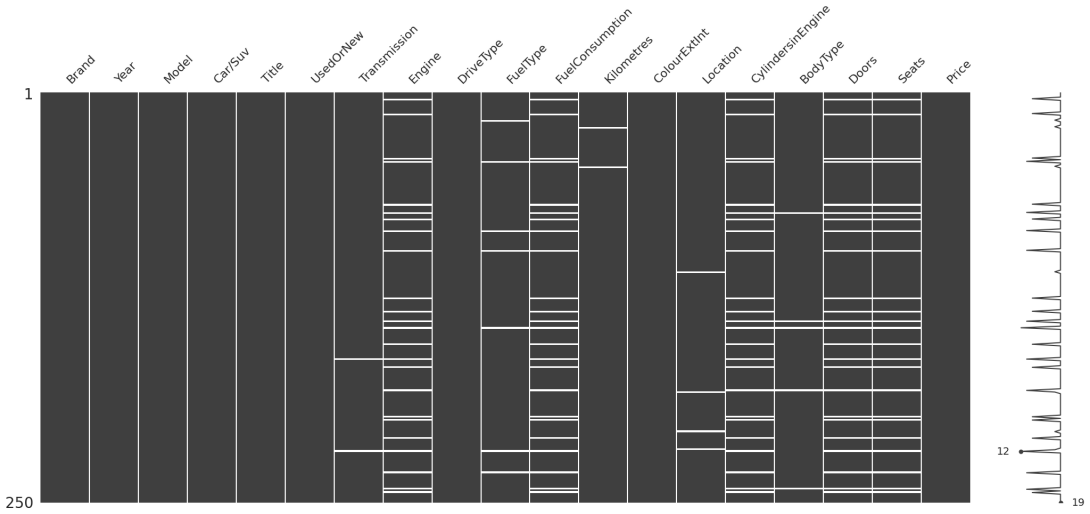
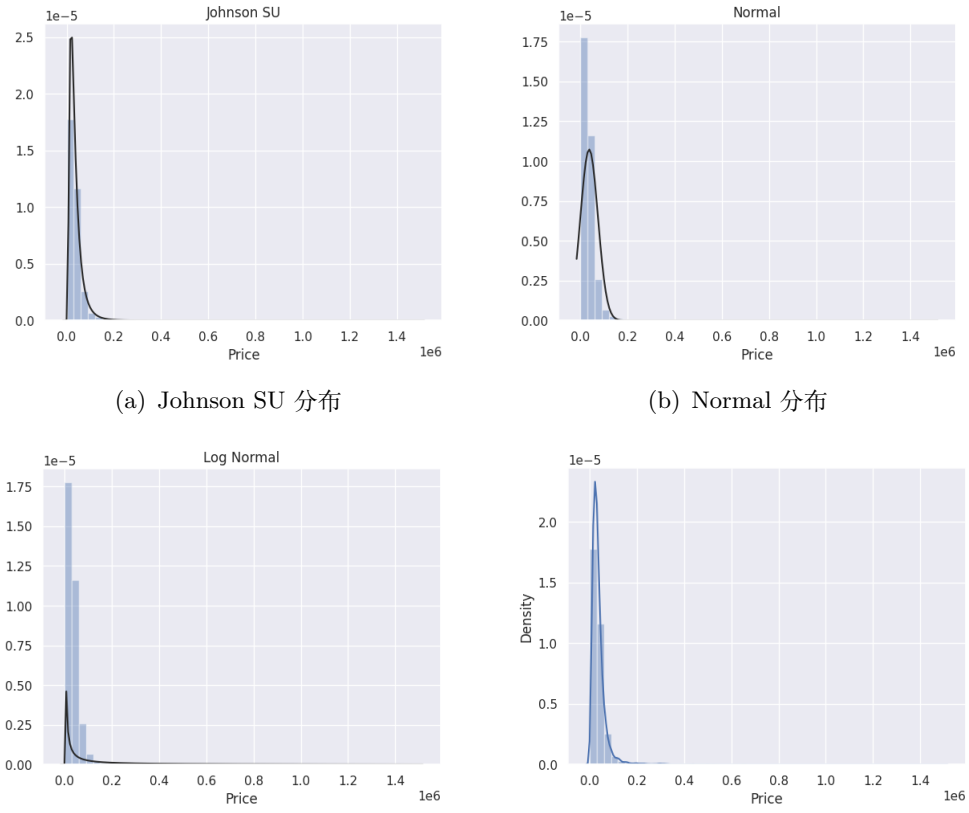


图 1.2: 缺失数据分布

对于缺失的数据，我们需要对其进行**合理的填充**，以增加数据量，保证模型的效果。同时，如果数据无法填充，则应该将其进行删除。我们还发现，“Seats” 和 “Doors” 两列数据有错位，部分”Seats” 数据被移至 “Doors” 的单元格。而在 “Price” 数据中，部分数值为 “POA”，查找资料可得知其含义为 “Price On Application”，这是无法作用于模型的训练和预测中的数据。

除此之外，我们发现除了”Year” 数据为”int64” 类型外，其余数据都是”object” 类型。为了在后续的程序中运用数据，我们需要将数据类型进行转换。

1.2 数据分析



(a) Johnson SU 分布

(b) Normal 分布

(c) Log Normal 分布

(d) 拟合后的 Johnson SU 分布

图 1.3: 价格分布图

在删除了值为 “POA” 的价格后，我们分析了价格的总体分布，如图1.3所示。最终我们发现，最佳拟合为**无界约翰分布**，如图1.3(a)和图1.3(d)所示。无界约翰分布有两个指标，即

- Skewness，即偏度，是述数据分布形状的统计量，其值为正表示数据分布偏向右侧（正偏），而其值为负表示数据分布偏向左侧（负偏）。在这里，偏度值为 8.660975，表明价格数据呈现明显的正偏分布。

- Kurtosis，即峰度，是描述数据分布尾部形状的统计量。正常的峰度值为 3，大于 3 表示尾部较重，小于 3 表示尾部较轻。在这里，峰度值为 190.370450，表明价格数据的分布尾部相当重，可能存在异常值或者极端的价格数据点。

我们绘制了数值型特征的热力图，如图1.4所示、各数值型特征的峰值和偏度如表??所示、数值型数据分布图如图1.5所示、数值型数据关系如图1.6所示。

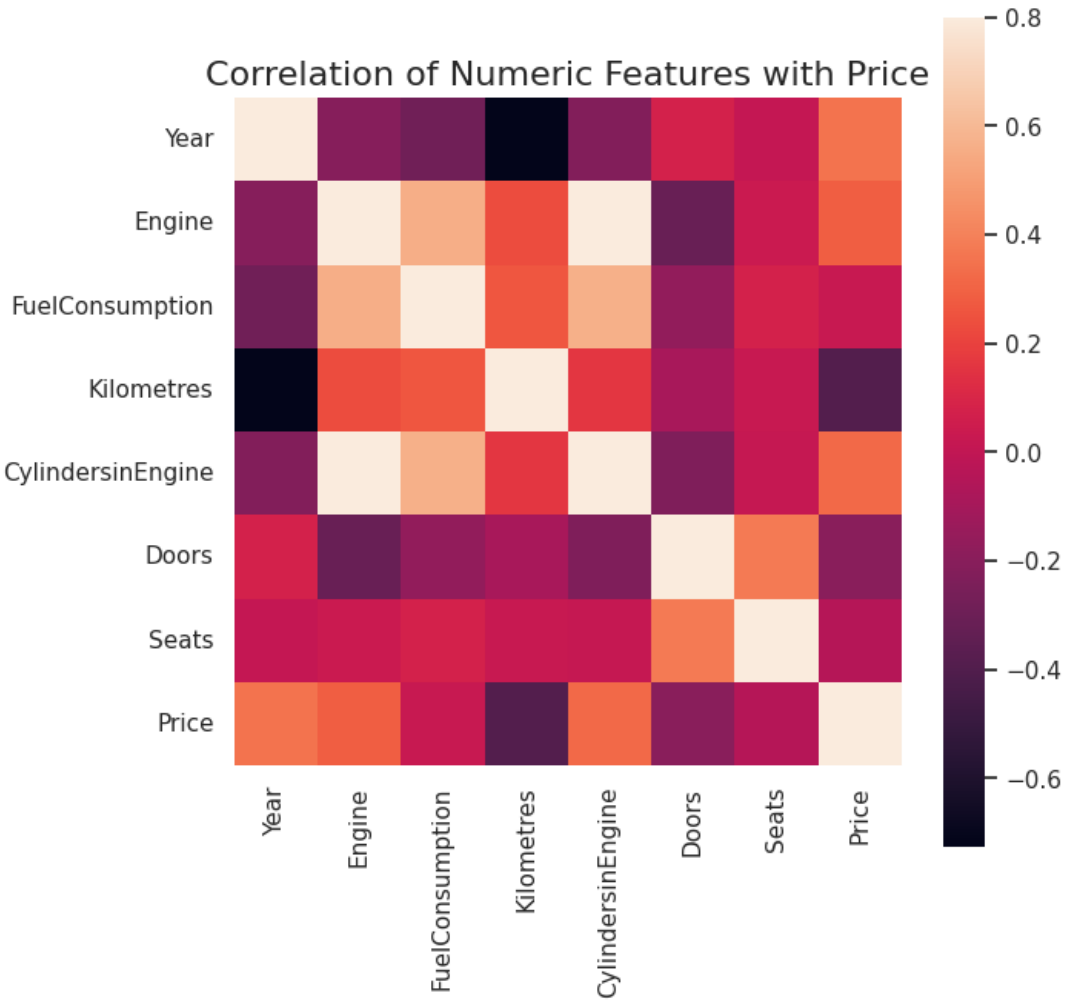


图 1.4: 数值型特征热力图

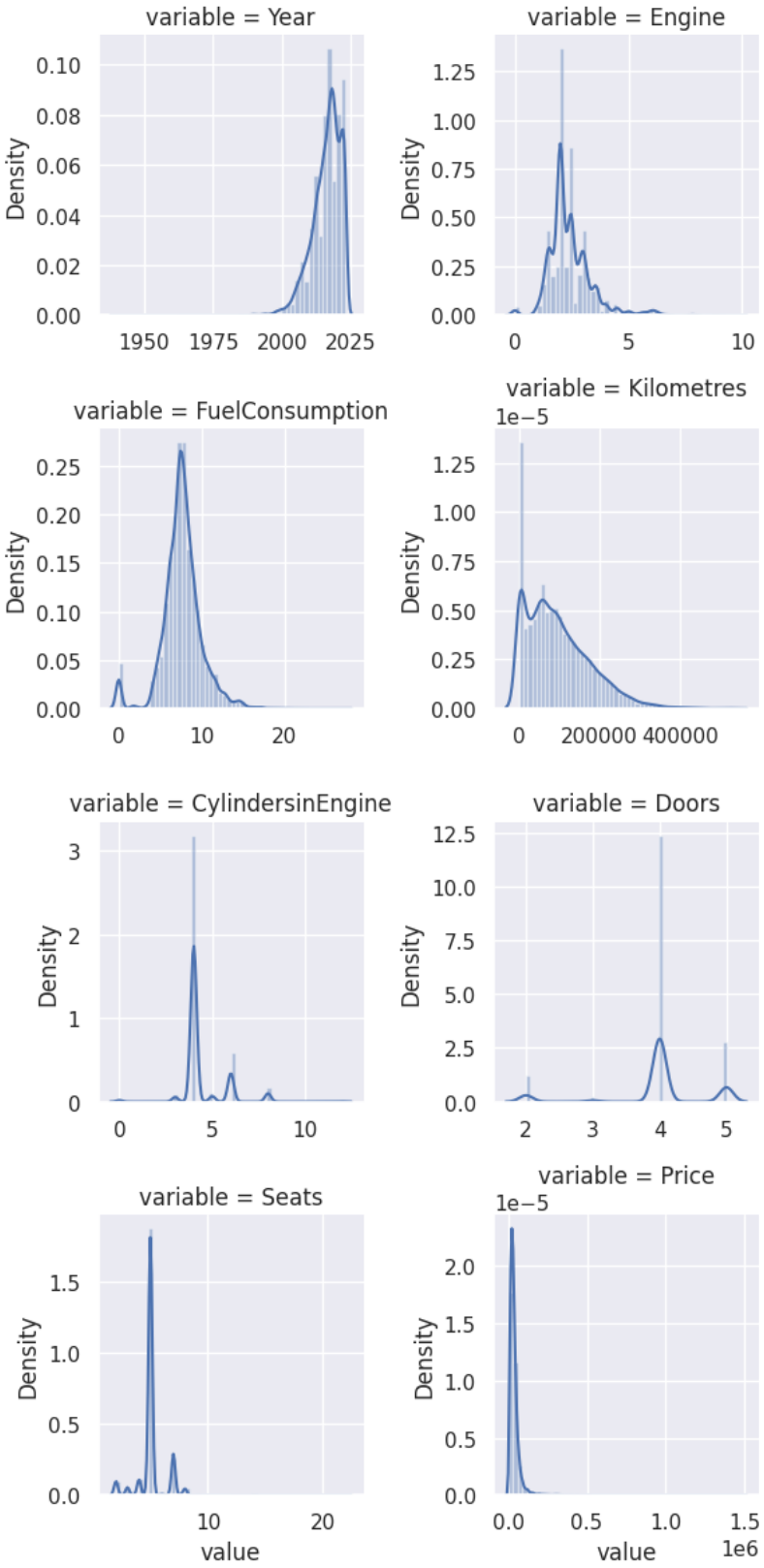


图 1.5: 数值型数据分布图

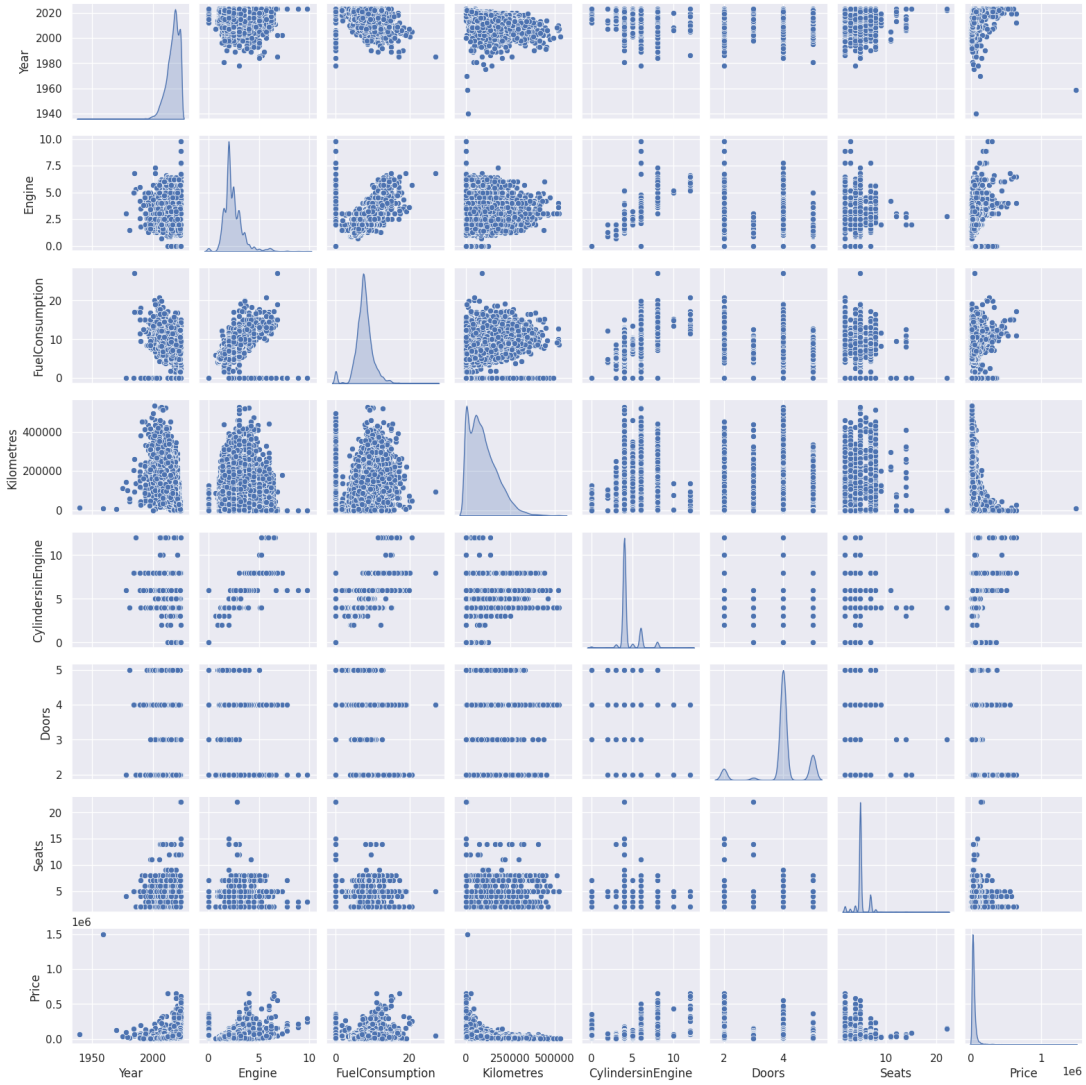


图 1.6: 数值型数据关系

对于离散型的数据，我们统计了部分数据的出现次数，如图1.7所示。

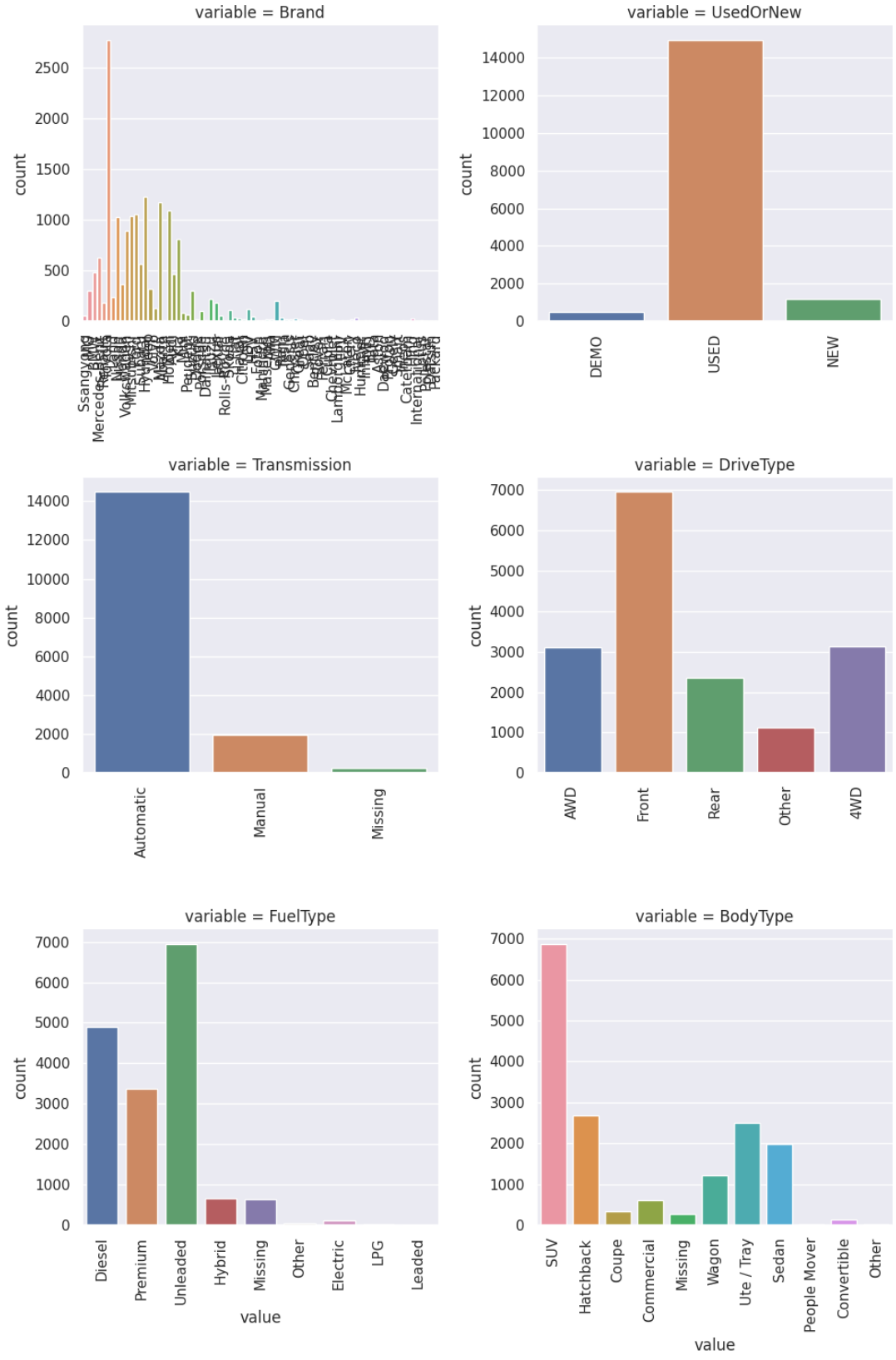

variable = Brand
count
variable = UsedOrNew
DEMO
USED
NEW
variable = Transmission
Automatic
Manual
Missing
variable = DriveType
AWD
Front
Rear
Other
4WD
variable = FuelType
Diesel
Premium
Unleaded
Hybrid
Missing
Other
Electric
LPG
Leaded
value
variable = BodyType
SUV
Hatchback
Coupe
Commercial
Missing
Wagon
Ute / Tray
Sedan
People Mover
Convertible
Other
value

图 1.7: 出现次数

1.3 数据填充

在所给出的数据中，有部分数据有缺失值。我们发现，对于已知 Title 的汽车，其发动机容量、油耗、发动机气缸数量、车门数量、座位数量是比较容易查找的数据。因此，我们可考虑从这几个方面入手。

我们利用 ChatGPT3.5 的 API，编写 Python 脚本（在附件的 car.ipynb 中可以找到源码），自动对数据进行填充。为了检验填充结果的正确性，我们手动查找了约 20 条汽车的资料，经过对比发现完全正确，因此可认为填充结果是可信的。

经过填充后，有缺失项的数据仅有三八多条，与原来的一千六百余条相比，大大降低了被损毁的数据。

1.4 特征工程

我们先对数据的异常值进行处理，利用箱型图进行删除，如图1.8所示。更多的箱型图可在 car.ipynb 文件中找到。

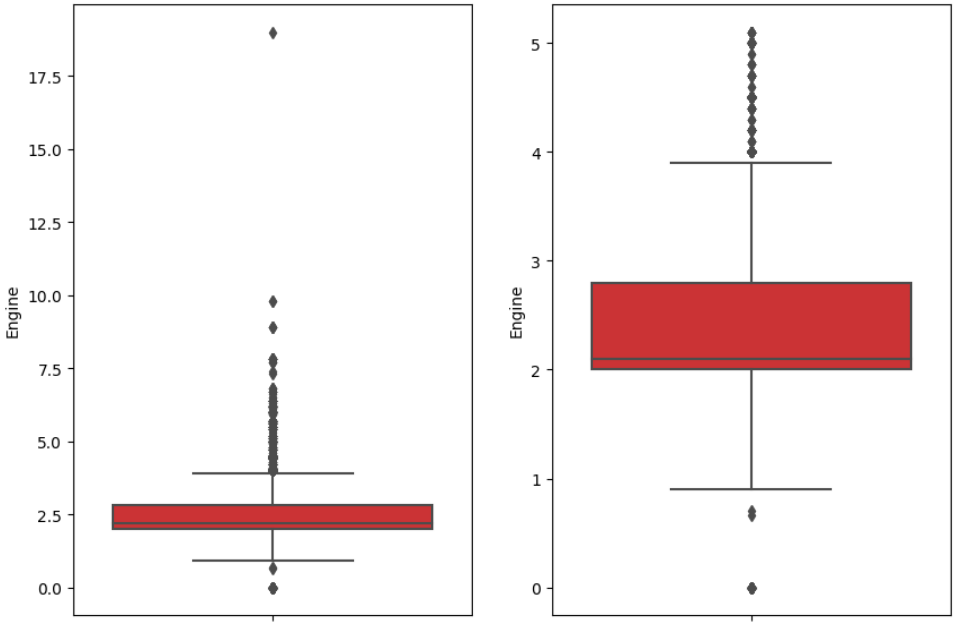


图 1.8: Engine 箱型图

除此之外，我们根据“Year”得到了汽车的出厂时长（定义为“Age”），因为我们认为汽车的出厂时长与汽车的价格关系较大。对于 Location，我们提取了其中州信息，抛弃了具体城市的信息。

由于离散型数据需要转化为数值型数据，而数值型数据是“偏序”的，即可比较大小。但是我们不希望离散型数据的大小是偏序的，因为部分数据并没有可比较性，只是不同的取值。因此，我们使用**独热编码**对其进行处理。

2 模型预测

我们使用了不同类型的模型进行预测，如 bp 神经网络、Linear Regression、Lasso、Ridge、随机森林回归、XGBRegression 等，最终发现 XGB 的效果最好，其 R2 达到了 **0.903**，MSE 为 71798280、RMSE 为 8473.4、MAE 为 4406.5。为以下是我们的实验过程。

2.1 bp 神经网络

我们将预处理后的数据分为训练集和测试集，直接将自变量和因变量放入 MLPClassifier 函数中进行预测，最终得到的 accuracy_score 极低（不足 0.01），且运算速度慢，直接抛弃该方案。

2.2 Linear Regression

我们将预处理后的数据删除掉 “ColourExtInt”、“Location”、“BodyType”、“Title” 后进行预测，并对离散数据进行独热编码，划分测试集和训练集，R2 能达到 0.7 左右。但是，我们认为 Title 对于数据预测具有一定作用，于是我们将出现频率较低的 Title 全部更改为 “Others”，保留出现频率高的 “Title”，并进行独热编码。最终，R2 降为负值，该方案被抛弃。但是更改后的 “Title” 在随机森林回归中发挥了作用。

2.3 Lasso

测试方案与 Linear Regression 类似，R2 略高于前者，能达到 0.73 左右。

2.4 Ridge

测试方案与 Linear Regression 类似，R2 略高于前者，能达到 0.76 左右。

2.5 随机森林回归

在直接使用预处理和独热编码后的数据进行预测时，R2 能达到 0.83 左右。但是，在我们使用改进的 “Title” 特征后，R2 可以达到 0.85 左右。

2.6 XGBRegression

我们发现，预测效果最好的模型是 XGBRegression。我们在预测之前，从 “Location” 中提取了州信息，抛弃了具体城市信息。同时，我们还抛弃了 Title、Model、Car/SUV、ColourExtInt、Brand 信息，并对剩余离散数据进行独热编码。此外，我们

构造了不同品牌的销售统计量作为新的特征。将 Engine，FuelConsumption，Kilometres,CylindersinEngine 以及销售统计量归一化后，便得到了训练所需自变量。

在训练之前，我们对 Price 取对数，作为因变量，即 $y = ln(\text{Price})$。在输出预测之后，通过反变换得到 $\hat{\text{Price}} = e^{\hat{y}}$。反变换后，计算 R2，即可得到 R2 为 0.903，是当前效果最好的方案。我们使用的具体参数为：

- n_estimators=100
- subsample=1.0
- learning_rate=0.22555555555555554
- booster = "gbtree"
- reg_lambda = 1.0 reg_alpha = 0.0
- max_depth = 6

3 其他数据分析

我们还分析了一些其他的数据，如不同年份生产的汽车销量。如图3.1所示，可以发现，总体上讲距今年份越近的汽车，销量越高。但是 2018 年生产的汽车，其销量达到了顶峰。

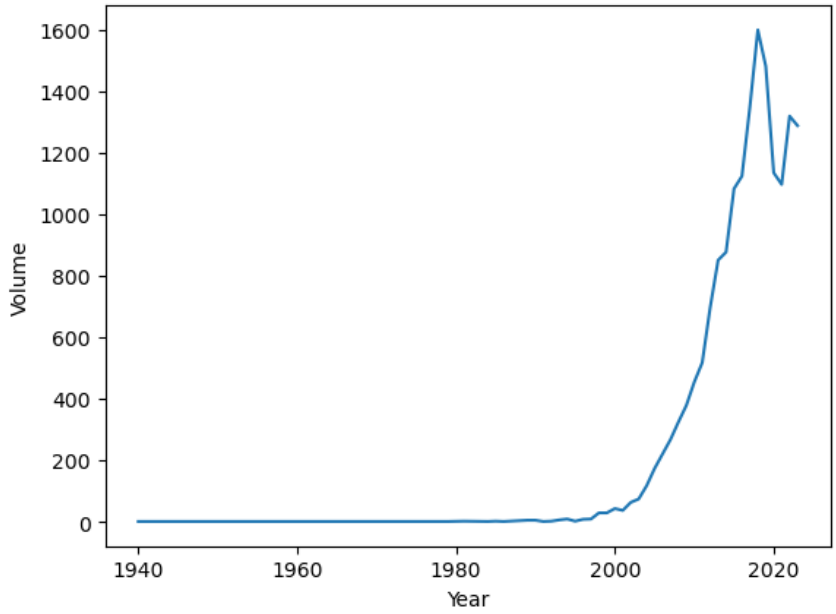


图 3.1: 不同年份生成汽车销售量

我们统计了不同州的销售占比，得到了如图3.2所示。可以看出 NSW 和 VC 两州的占比极大，超过一半的汽车都在这两个州销售，是主要市场。此外，我们统计了不同燃料汽车的销售情况，如图3.3所示。可以发现，在 2020 年前后，燃料类型为 Diesel 的汽车销量大减，但是 Electric 和 Hybrid 的销量都明显增加，可以发现这两者抢占了市场份额。但在此之后 Diesel 的销量有所回升，因此可以预测，在很长一段时间内，Diesel 的销量还是会占据主要市场份额，Electric 与 Hybrid 类型的汽车销量也会趋于平稳。

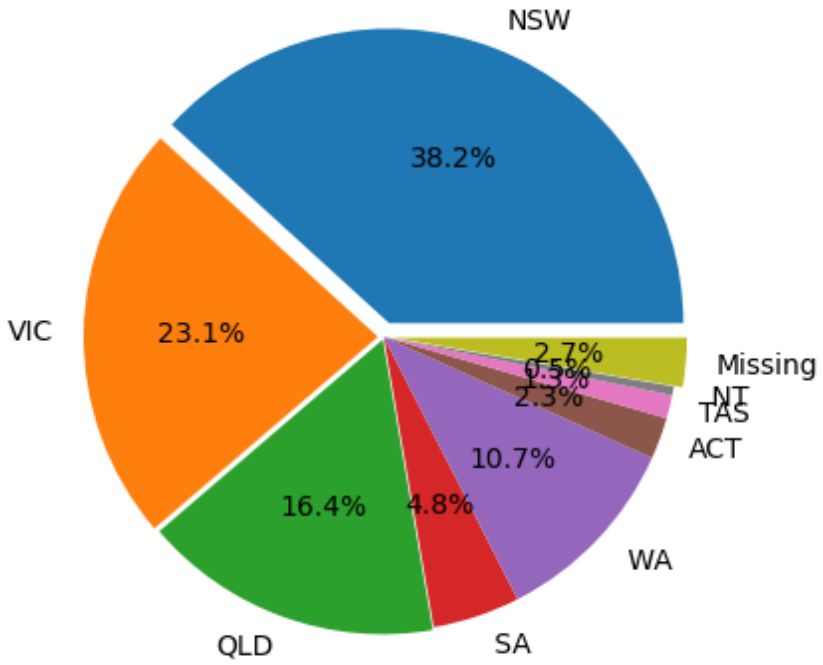


图 3.2: 各州销售占比

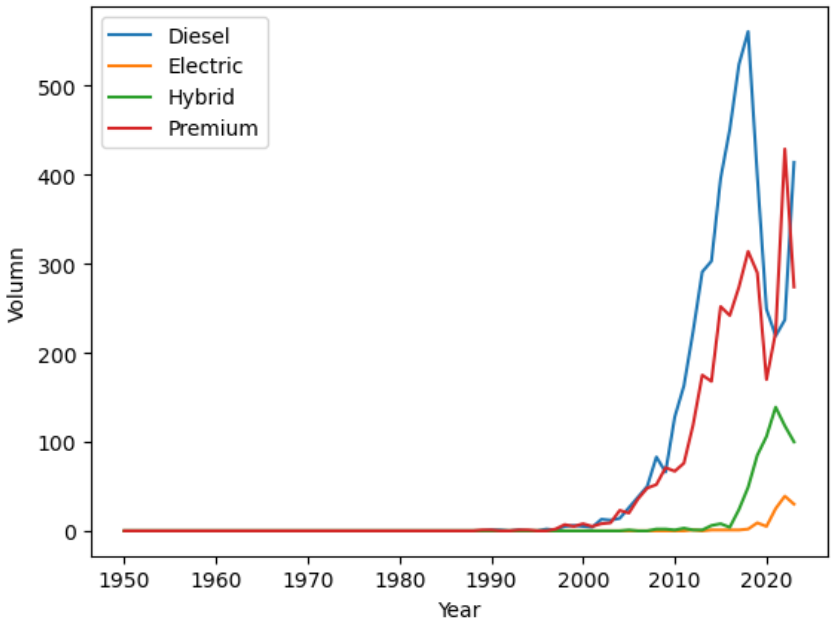


图 3.3: 不同燃料类型数据