丰田卡罗拉汽车二手回收价格预测分析报告

目录

1.摘要

本案例从 kaggle 社区获取了一家大型丰田汽车经销商回收二手卡罗拉的价格和汽车情况的数据集。并根据数据集对数据进行了清洗筛选，通过随机森林模型对六个自变量和一个因变量进行了建模预测，以期望为经销商和顾客提供回收价格预测参考。

2．数据

2.1 数据来源及描述

一家大型的丰田汽车经销商为购买新款丰田汽车的客户提供了一项特殊的置换选择。具体而言，一项新的促销活动承诺为购买新车的客户提供较高的价格，以将他们的旧丰田卡罗拉汽车置换掉。随后，经销商会以微薄的利润将这些旧车重新售出。为了确保经销商能够获得合理的利润，他们需要能够准确预测旧车的销售价格。因此，他们收集了有关该经销商以前所有销售的旧丰田卡罗拉汽车的数据。这些数据包括车辆的销售价格以及其他相关信息，如车龄、行驶里程等。

数据来源于 Kaggle 社区中 Price of Used Toyota Corolla Cars 数据集，数据集中的记录总数为 1436 辆汽车。每个变量的描述如下：

Id：每个汽车条目的唯一标识符。

Model：丰田卡罗拉的车型名称。

Price：二手车的售价。

Age_08_04：截至 2004 年 8 月的汽车车龄（以月份为单位）。

Mfg_Year：汽车的制造年份。

KM：汽车行驶的公里数。

Fuel_type：汽车使用的燃料类型（例如汽油、柴油）。

HP：马力，衡量发动机功率的指标。

Met_Color：金属漆的二进制指示器（0 = 否，1 = 是）。

Color：汽车的外观颜色。

Automatic：自动变速器的二进制指示器（0 = 否，1 = 是）。

CC：发动机尺寸（以立方厘米为单位）。

Doors：车上的车门数量。

Quarterly：汽车的季度税。

Weight：汽车的重量。

Mfr_Guarantee：制造商保证的二进制指标（0 = 否，1 = 是）。

BOVAG_Guarantee：BOVAG 保证的二进制指标（0 = 否，1 = 是）。

Guarantee_Period：保修期限（以月为单位）。

ABS：防抱死制动系统的二进制指示器（0 = 否，1 = 是）。

Airco：空调的二进制指示器（0 = 否，1 = 是）。

Automatic_airco：自动空调的二进制指示器（0 = 否，1 = 是）。

Boardcomputer：车机的二进制指示器（0 = 否，1 = 是）。

CD_Player：CD 播放器存在的二进制指示器（0 = 否，1 = 是）。

Central_Lock：中控锁系统的二进制指示器（0 = 否，1 = 是）。

Powered_Windows：通电窗口的二进制指示器（0 = 否，1 = 是）。

Radio：收音机的二进制指示器（0 = 否，1 = 是）。

Mistlamps：雾灯的二进制指示器（0 = 否，1 = 是）。

Sport_Model：运动模型的二进制指标（0 = 否，1 = 是）。

Backseat_Divider：后座隔板的二进制指示器（0 = 否，1 = 是）。

Metallic_Rim：金属轮辋的二进制指示器（0 = 否，1 = 是）。

2.2 数据预处理与分析过程

2.2.1 数据预处理

数据描述：

使用 info()，describe()，方法对数据进行描述，得知数据中并不存在空值，并得知每种指标的统计性分析，如下图所示。

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1436 entries, 0 to 1435
Data columns (total 39 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Id                 1436 non-null   int64
 1   Model              1436 non-null   object
 2   Price              1436 non-null   int64
 3   Age_08_04          1436 non-null   int64
 4   Mfg_Month          1436 non-null   int64
 5   Mfg_Year           1436 non-null   int64
 6   KM                 1436 non-null   int64
 7   Fuel_Type          1436 non-null   object
 8   HP                 1436 non-null   int64
 9   Met_Color          1436 non-null   int64
 10  Color              1436 non-null   object
 11  Automatic          1436 non-null   int64
 12  CC                 1436 non-null   int64
 13  Doors              1436 non-null   int64
 14  Cylinders          1436 non-null   int64
 15  Gears              1436 non-null   int64
 16  Quarterly_Tax      1436 non-null   int64
 17  Weight             1436 non-null   int64
 18  Mfr_Guarantee      1436 non-null   int64
 19  BOVAG_Guarantee    1436 non-null   int64
 20  Guarantee_Period   1436 non-null   int64
 21  ABS                1436 non-null   int64
 22  Airbag_1           1436 non-null   int64
 23  Airbag_2           1436 non-null   int64
 24  Airco              1436 non-null   int64
 25  Automatic_airco    1436 non-null   int64
 26  Boardcomputer      1436 non-null   int64
 27  CD_Player          1436 non-null   int64
 28  Central_Lock       1436 non-null   int64
 29  Powered_Windows    1436 non-null   int64
 30  Power_Steering     1436 non-null   int64
 31  Radio              1436 non-null   int64
 32  Mistlamps          1436 non-null   int64
 33  Sport_Model        1436 non-null   int64
 34  Backseat_Divider   1436 non-null   int64
 35  Metallic_Rim       1436 non-null   int64
 36  Radio_cassette     1436 non-null   int64
 37  Parking_Assistant  1436 non-null   int64
 38  Tow_Bar            1436 non-null   int64
dtypes: int64(36), object(3)
memory usage: 437.7+ KB
```

图 2-1 关于价格的相关系数热力图

	Id	Price	Age_08_04	Mfg_Month	Mfg_Year	KM	HP	Met_Color	Automatic	CC	...	Powered_Win
count	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.000000	1436.00000	...	1436.0
mean	721.555014	10730.824513	55.947075	5.548747	1999.625348	68533.259749	101.502089	0.674791	0.055710	1576.85585	...	0.5
std	416.476890	3626.964585	18.599988	3.354085	1.540722	37506.448872	14.981080	0.468616	0.229441	424.38677	...	0.4
min	1.000000	4350.000000	1.000000	1.000000	1998.000000	1.000000	69.000000	0.000000	0.000000	1300.00000	...	0.0
25%	361.750000	8450.000000	44.000000	3.000000	1998.000000	43000.000000	90.000000	0.000000	0.000000	1400.00000	...	0.0
50%	721.500000	9900.000000	61.000000	5.000000	1999.000000	63389.500000	110.000000	1.000000	0.000000	1600.00000	...	1.0
75%	1081.250000	11950.000000	70.000000	8.000000	2001.000000	87020.750000	110.000000	1.000000	0.000000	1600.00000	...	1.0
max	1442.000000	32500.000000	80.000000	12.000000	2004.000000	243000.000000	192.000000	1.000000	1.000000	16000.00000	...	1.0

8 rows × 36 columns

图 2-2 关于价格的相关系数热力图

相关性分析：

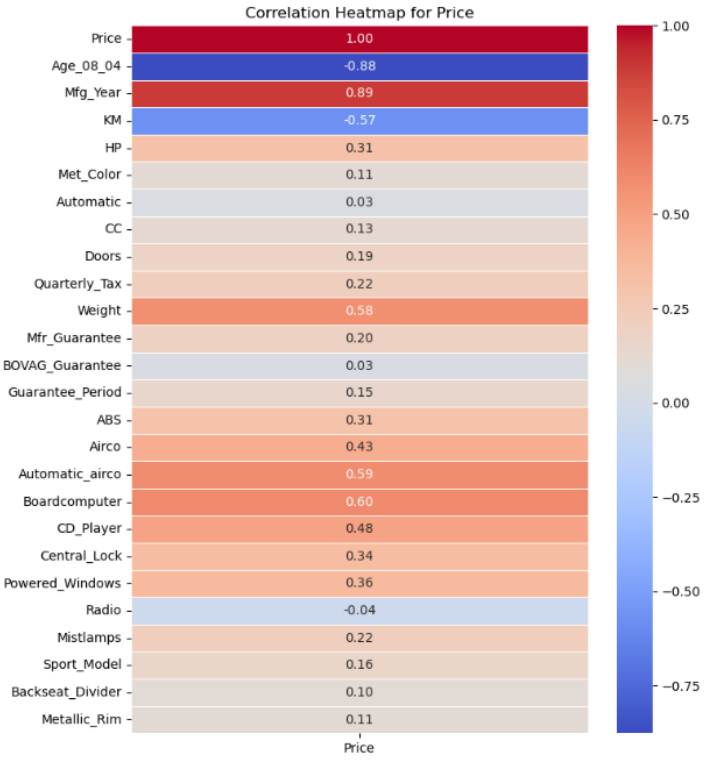

图 2-3 关于价格的相关系数热力图

根据已知数据构建关于价格的相关系数矩阵并绘制相关系数热力图。通过热力图观察排除相关性过低的指标，本案例以 0.5 为标准进行筛选，最终得到 Age_08_04，Mfg_Year，KM，Weight，Automatic_airco，Boardcomputer 六个指标，并进行筛选构建新的数据表。

描述分析：

（一）因变量——Price：二手车的售价。

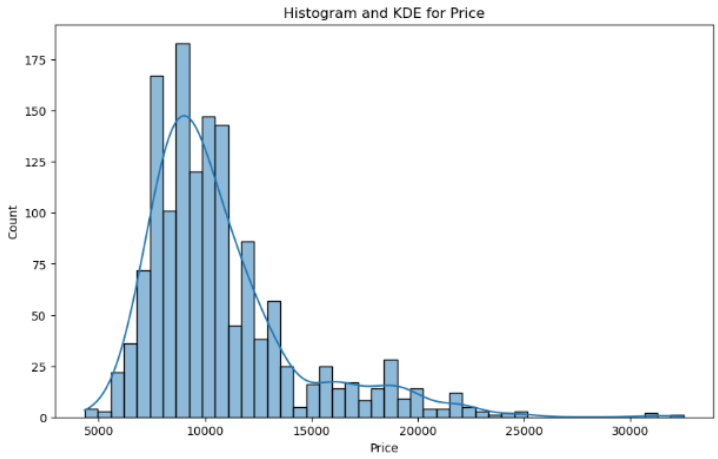

图 2-4 价格直方图

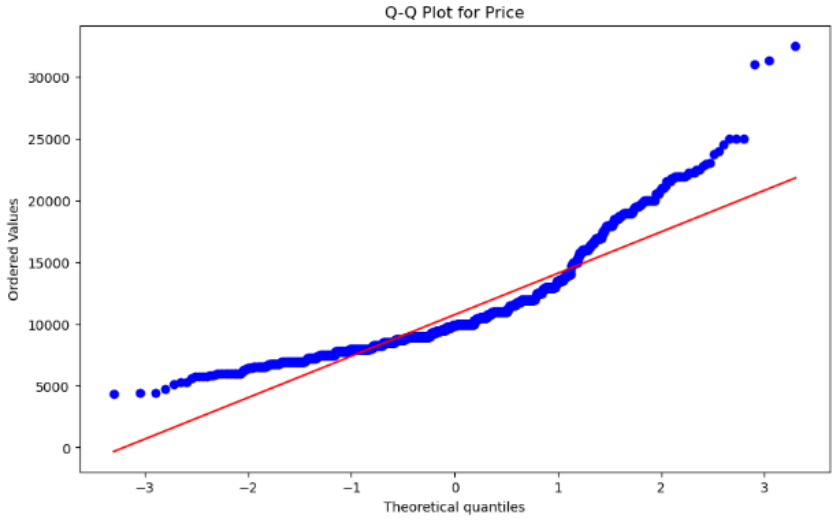

图 2-5 价格正态概率图

通过对二手车售价的直方图的观察可以看出，二手车售价的数据大致呈现正态分布，平均数为 10730.824513 美元，中位数为 9900 美元，数据总量为 1436，最小值为 4350 美元，最大值为 32500 美元，本案例将其中大于 30000 的值定义为异常值，并将其删除。

（二）自变量——Age_08_04：截至 2004 年 8 月的汽车车龄（以月份为单位）。

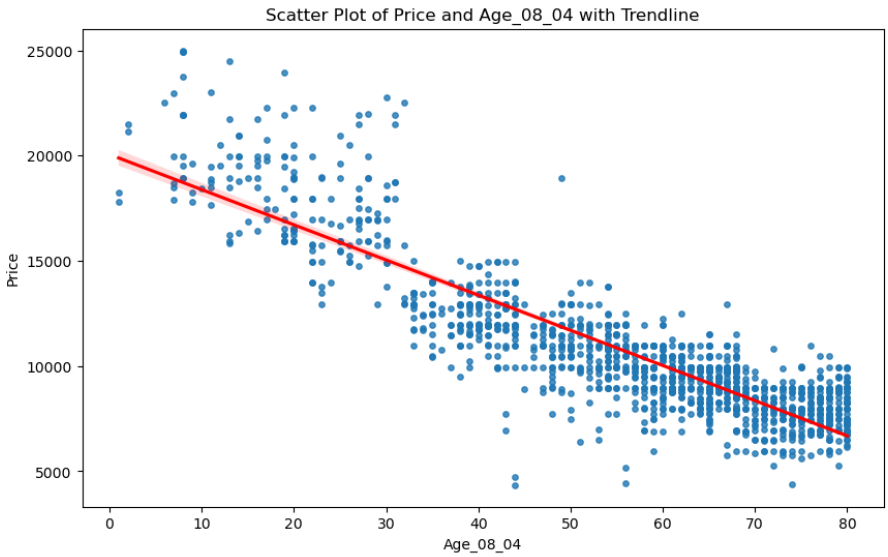

图 2-6 价格与车龄散点图

根据散点图和趋势线可以看出，价格与汽车车龄之间存在明显负相关关系，相关系数为-0.88，呈现显著负相关，价格随着汽车车龄的增大而减少。

（三）自变量——Mfg_Year：汽车的制造年份。

价格与汽车车龄之间存在明显正相关关系，相关系数为 0.89，呈现显著正相关，汽车生产时间越大，回收价格也就更高，与汽车车龄同理。

（四）自变量——KM：汽车行驶的公里数。

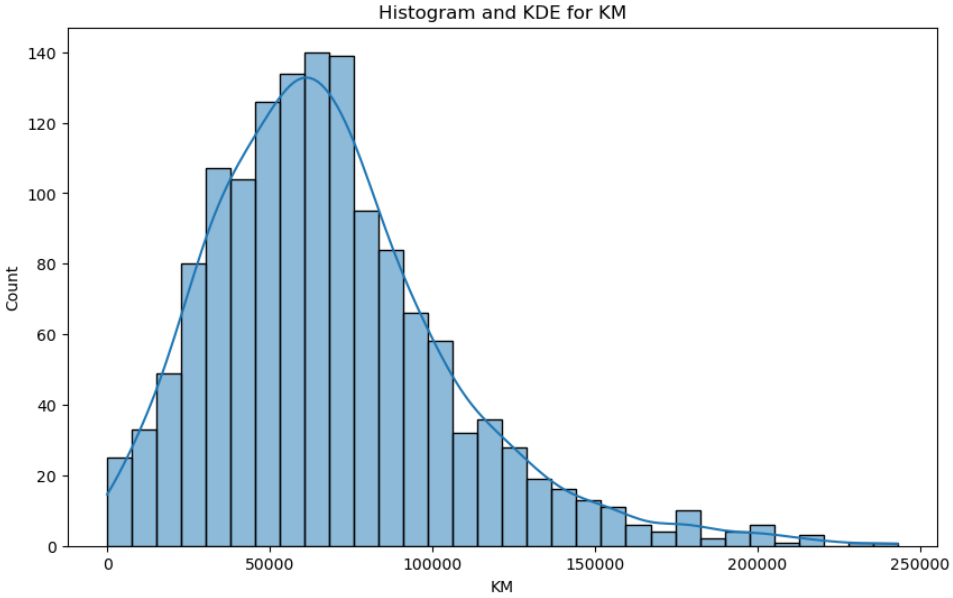

图 2-7 KM 直方图

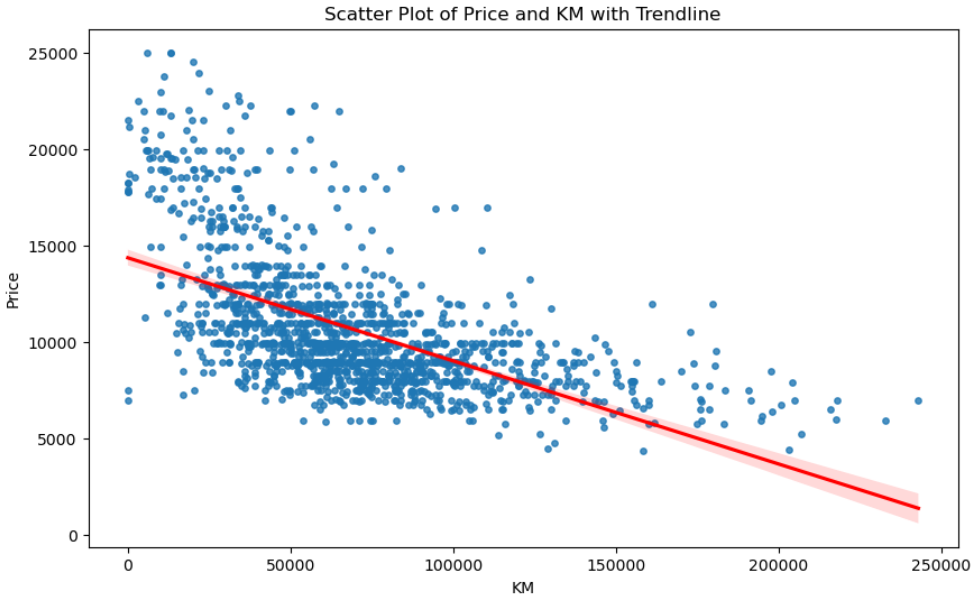

图 2-8 KM 与价格散点图

根据 KM 绘制的直方图表明，KM 数据比较符合正态分布，且数据较为集中。

KM 与价格之间存在负相关关系，负相关系数为-0.57，二手回收价格会随着汽车行驶公里数的增加而减少。

（五）自变量——Weight：汽车的重量。

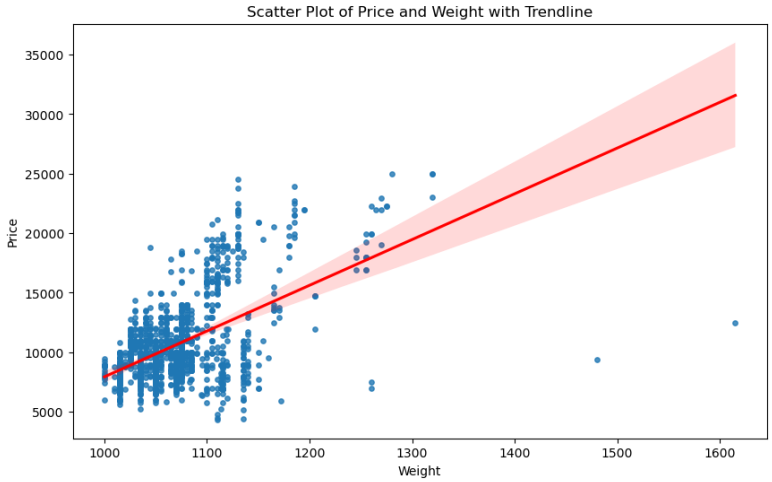

图 2-9 重量与价格散点图

根据重量与价格的散点图可以看出，重量与价格之间存在正相关关系，相关性系数为 0.58，汽车的重量往往代表其中重要零件是否得以保存，所以回收时汽车的重量越大，回收价格可能越高。

（六）自变量——Automatic_airco：自动空调的二进制指示器（0 = 否，1 = 是）。

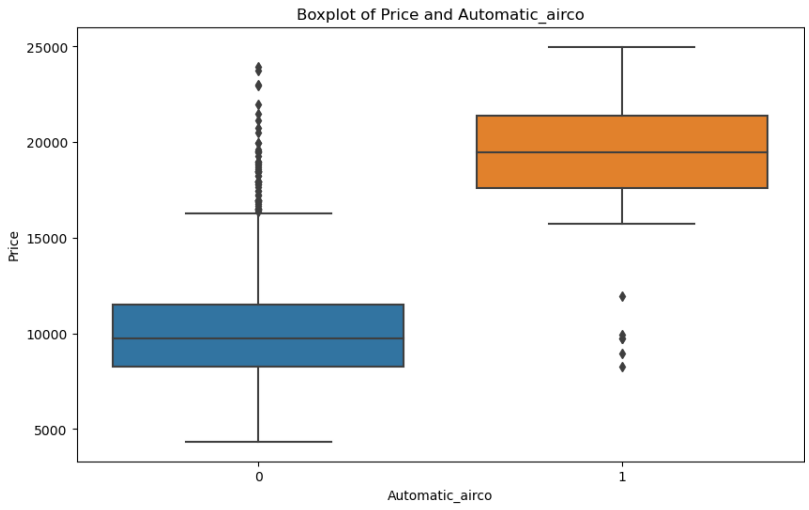

图 2-10 自动空调与价格箱线图

根据自动空调与价格的箱线图可以看出，带有自动空调的二手卡罗拉往往能获得更高的回收价格，在 2002 至 2004 年的时间段，自动空调往往是高价值选配功能，故而能获得更高价格。

（七）自变量——Boardcomputer：车机的二进制指示器（0 = 否，1 = 是）。

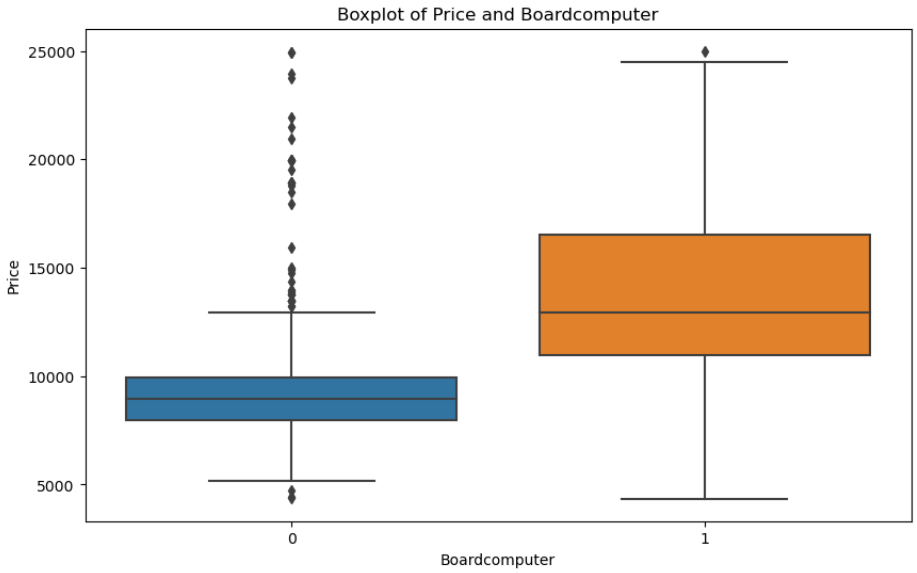

图 2-11 车机与价格箱线图

根据车机与价格箱线图可以看出，带有车机的二手卡罗拉往往比没有车机的二手卡罗拉能获得更高的回收价格，与自动空调同理，车机在 2002 至 2004 年期间是比较高端的选配功能，故而又较高回收价值。

2.2.2 数据分析过程

本案例通过筛选出的相关性较高的六个自变量，以价格为因变量构建随机森林模型，并绘制自变量的重要性柱状图，模型得到的 R^2 为 0.889，(MSE): 1207243.346，(MAE): 835.500，(EVS): 0.889。表明模型有较高拟合程度，使用 GridSearchCV 进行超参数搜索对模型进行优化后，得到的 n_estimators 参数为 200，R^2 为 0.891，相比起优化之前提升 0.002。重要性排序柱状图如下，如图所示，从图表中可以看出，汽车车龄和制造年份为最重要的自变量；公里数，车机，重量，自动空调相比前两者而言没那么重要，商家可以根据以下指标的重要性对车辆价值进行预测，根据随机森林模型的预测结果以新一列添加到数据中，结果如下表所示。

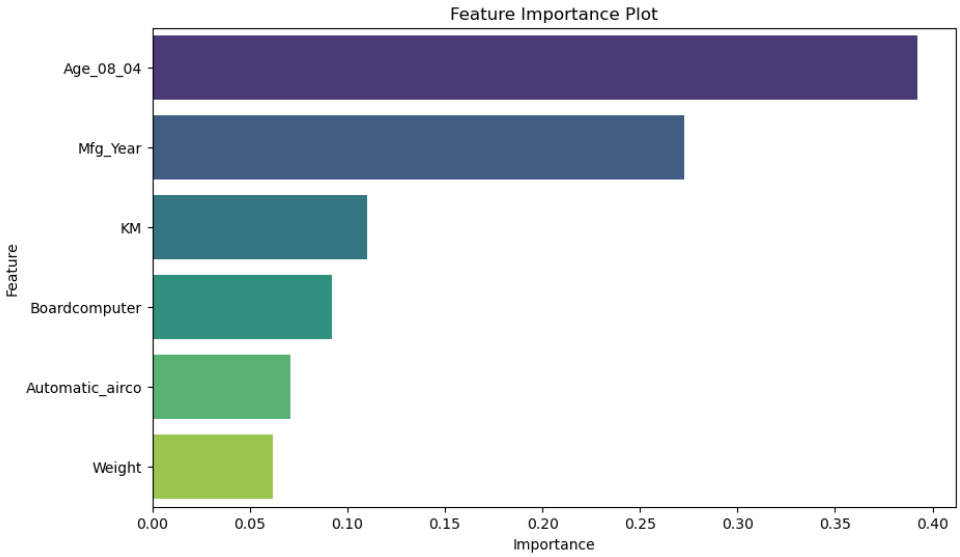

图 2-8 随机森林模型自变量重要性排序

表 2-1 最终预测值前五行

	Price	Age_08_04	Mfg_Year	KM	Weight	Automatic_airco	Boardcomputer	Predicted_Price
0	13500	23	2002	46986	1165	0	1	13989.055102
1	13750	23	2002	72937	1165	0	1	14121.752820
2	13950	24	2002	41711	1165	0	1	14319.752434
3	14950	26	2002	48000	1165	0	1	14962.795758
4	13750	30	2002	38500	1170	0	1	14542.110504

3.分析结果

在二手卡罗拉回收的过程中，回收价格随着汽车车龄的增加而降低，制造年份越大的汽车，获得更高回收价格的可能性更高。行驶公里数代表了汽车的使用强度，长距离的行驶会导致汽车零件的老化磨损，降低回收时的价格。车机和自动空调在 2002 至 2004 年时属于高价值选配功能，故而在回收时如果汽车中存在这两种功能更有可能获得更高的收购价格。汽车重量也一定程度上表示了汽车重要配件是否齐全，更接近标准重量的汽车更有可能获得更高的回收价格。

4.结论与展望

综上所述，本案例构建的随机森林模型有较高拟合度，对模型进行检验后表明模型有较高精确度。根据随机森林模型绘制的特征重要性柱状图表示，汽车车龄，制造年份是影响二手车回收价格的最重要的两个因素，行驶公里数，是否有车机和自动空调，汽车重量也对二手车回收价格存在不同程度的影响。

通过本案例的分析，卖家和买家在考虑二手卡罗拉回收时能更直观的发现比较重要的回收价格影响因素。从而做出更合理的回收价格预测。卖家如果想要获得更高的回收价格，可以考虑好购买和回收的年份间隔，合理规划时间以获得更高收益。控制行驶公里数，减少零件的磨损老化也可能获得更高的回收价格。在购买时选配高价值的功能可能在回收时有机会获得更高的回收价格。

5.1 附表

图 5-1 随机森林代码

In [25]:
```
# 选择自变量和因变量
features = ['Age_08_04', 'Mfg_Year', 'KM', 'Weight', 'Automatic_airco', 'Boardcomputer']
target = 'Price'
```

In [26]:
```
# 提取特征和目标变量
X = corolla[features]
y = corolla[target]
```

In [45]:
```
# 划分数据集为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 创建随机森林回归模型
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

# 训练模型
rf_model.fit(X_train, y_train)
```

Out[45]:
```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [46]:
```
# 在测试集上进行预测
y_pred = rf_model.predict(X_test)
```

In [47]:
```
# 评估模型性能
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
evs = explained_variance_score(y_test, y_pred)
max_err = max_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error (MSE): {mse}')
print(f'Mean Absolute Error (MAE): {mae}')
print(f'Explained Variance Score (EVS): {evs}')
print(f'Maximum Error: {max_err}')
print(f'R-squared (R2): {r2}')
```

```
Mean Squared Error (MSE): 1207243.3460943862
Mean Absolute Error (MAE): 835.5004645760745
Explained Variance Score (EVS): 0.8897255704041869
Maximum Error: 4347.6
R-squared (R2): 0.8896988799480082
```

In [48]:
```
# 预测新样本
new_data = np.array([[3, 2002, 80000, 1200, 1, 1]])  # 请根据实际情况提供新样本的特征
predicted_price = rf_model.predict(new_data)
print(f'Predicted Price for the new sample: {predicted_price}')
```

```
Predicted Price for the new sample: [19321.2]
```

图 5-2 参数优化代码

In [74]:

```
#优化参数
# 定义参数网格
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt', 'log2'],
    'min_samples_split': [2, 5, 10]
}

# 创建随机森林回归模型
rf_model = RandomForestRegressor(random_state=42)

# 使用 GridSearchCV 进行超参数搜索
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5, scoring='neg_
grid_search.fit(X_train, y_train)

# 输出最佳参数
best_params = grid_search.best_params_
print(f'Best Parameters: {best_params}')

# 使用最佳参数的模型进行预测
best_rf_model = grid_search.best_estimator_
y_pred_optimized = best_rf_model.predict(X_test)

# 评估优化后模型的性能
mse_optimized = mean_squared_error(y_test, y_pred_optimized)
print(f'Mean Squared Error (Optimized): {mse_optimized}')

mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
evs = explained_variance_score(y_test, y_pred)
max_err = max_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error (MSE): {mse}')
print(f'Mean Absolute Error (MAE): {mae}')
print(f'Explained Variance Score (EVS): {evs}')
print(f'Maximum Error: {max_err}')
print(f'R-squared (R2): {r2}')

# 预测新样本
new_data = np.array([[3, 2002, 80000, 1200, 1, 1]])  # 请根据实际情况提供新样本的特征
predicted_price_optimized = best_rf_model.predict(new_data)
print(f'Predicted Price for the new sample (Optimized): {predicted_price_optimized}')
```

图 5-3 优化结果代码

In [25]:
```
# 选择自变量和因变量
features = ['Age_08_04', 'Mfg_Year', 'KM', 'Weight', 'Automatic_airco', 'Boardcomputer']
target = 'Price'
```

In [26]:
```
# 提取特征和目标变量
X = corolla[features]
y = corolla[target]
```

In [42]:
```
# 划分数据集为训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 创建随机森林回归模型
rf_model = RandomForestRegressor(n_estimators=200, random_state=42)

# 训练模型
rf_model.fit(X_train, y_train)
```

Out[42]:
```
RandomForestRegressor
RandomForestRegressor(n_estimators=200, random_state=42)
```

In [43]:
```
# 在测试集上进行预测
y_pred = rf_model.predict(X_test)
```

In [44]:
```
# 评估模型性能
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
evs = explained_variance_score(y_test, y_pred)
max_err = max_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error (MSE): {mse}')
print(f'Mean Absolute Error (MAE): {mae}')
print(f'Explained Variance Score (EVS): {evs}')
print(f'Maximum Error: {max_err}')
print(f'R-squared (R2): {r2}')
```

```
Mean Squared Error (MSE): 1188294.7993971833
Mean Absolute Error (MAE): 829.4771022067363
Explained Variance Score (EVS): 0.8914531686592941
Maximum Error: 4063.2250000000004
R-squared (R2): 0.8914301348195472
```

In [35]:
```
# 预测新样本
new_data = np.array([[3, 2002, 80000, 1200, 1, 1]])  # 请根据实际情况提供新样本的特征
predicted_price = rf_model.predict(new_data)
print(f'Predicted Price for the new sample: {predicted_price}')
```

```
Predicted Price for the new sample: [19321.2]
```