

Critical Analysis Report on the Data Science Project Based on Australian Vehicle Prices Dataset

Prepared by:

Michelle David

47911891

Submitted to Macquarie University

Table of Contents

Introduction	3
Executive Summary	3
Critical Issue 1: Visualization not aligned with analysis goal	4
Problem Identification	4
Justification.....	4
Correction	5
Critical Issue 2: Feature Scaling	5
Problem Identification	5
Justification.....	6
Correction	6
Critical Issue 3: Poor Feature Selection Method	7
Problem Identification	7
Justification.....	8
Correction	8
Critical Issue 4: Incorrect and Incomplete Analysis of Results	9
Problem Identification	9
Justification.....	9
Correction	10

Introduction

This report analyses the “Data Science Project Based on Australian Vehicle Prices Dataset” focusing on the critical issues in the implementation of the project. This includes comparing the performance of the different models used for predicting car prices, namely, Linear Regression (LR), Decision Tree Regressor (DTR), and Multi-Layer Perceptron (MLP). This report looks into the issues and proposes ways to improve the implementation of the project while ensuring it aligns with the projects goal of (1) predicting car prices based on their features and (2) comparing the performances of the different models used.

Executive Summary

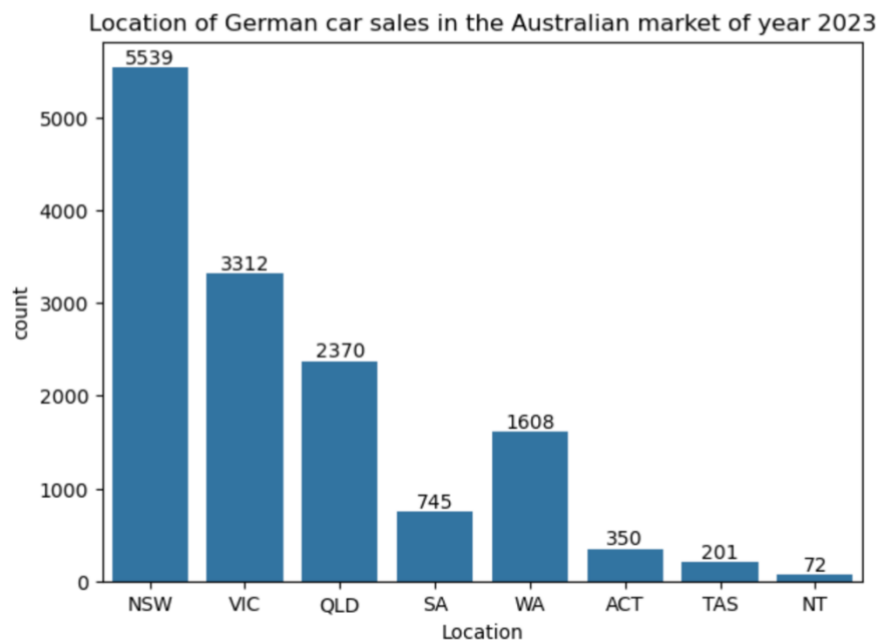
This report carefully addresses four critical issues affecting the precision of the models in predicting cars' prices. To begin with, the utilization of a count plot of car sales by location did not align with the project's aim, hence it was substituted with a box plot of prices based on car's transmission type to provide a better insight of the dataset. Subsequently, the lack of feature scaling after encoding can weaken MLP model's performance, so it was resolved by implementing StandardScaler. Moreover, the basic approach to feature selection solely based on correlation was revised, replacing it with Recursive Feature Elimination (RFE) to capture more complex relationships. Finally, incorporating Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2) values into the analysis of results offers a complete and informative assessment of the models' performance. The revised analysis concludes that all models performed just moderately, highlighting the need for more model tuning and data preparation to improve predictive precision, while also considers exploring other advanced models that can better capture complex relationships present in the dataset.

Critical Issue 1: Visualization not aligned with analysis goal

Problem Identification

The initial plot, showing count plot of car sales by location does not align with the goal of predicting car prices using features such as transmission type, production year, etc. The visual representation sheds light on how sales are spread out across locations but is not connected in predicting car prices.

```
1: plt.figure(figsize=(7,5))
   ax = sns.countplot(data=data, x='Location')
   for i in ax.containers:
       ax.bar_label(i,)
   plt.title('Location of German car sales in the Australian market of year 2023')
   plt.show()
```



Justification

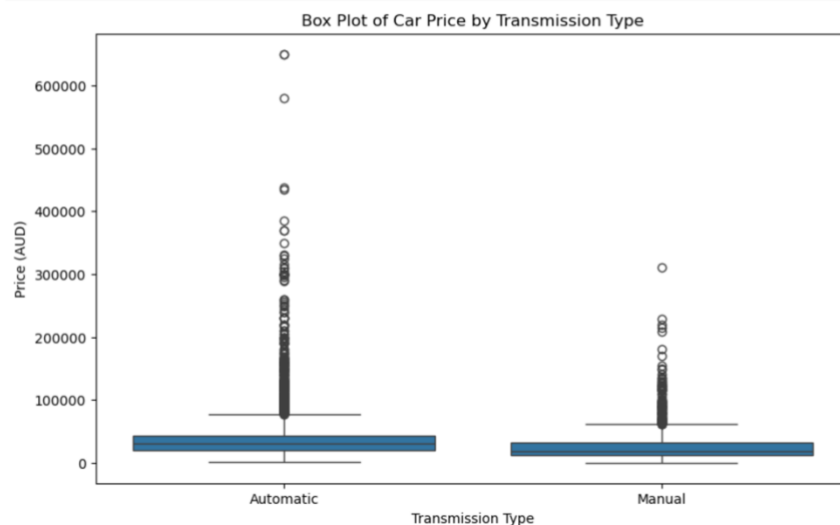
Comparing the prices of cars based on their transmission type using a box plot is a better way to explore the relationships related to price. The plot offers valuable insight into how the type of

transmission could impact the pricing of vehicles. This supports the goal of comprehending how different features affect pricing and assists in building the prediction models.

Correction

To address this issue, a new box plot is generated to showcase price fluctuations between automatic and manual transmissions which assist in spotting any patterns or notable price differences that could enhance the precision of the models. This adjustment brings the analysis more in line with its intended goal by connecting price directly to car features.

```
[45]: plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='Transmission', y='Price')
plt.title('Box Plot of Car Price by Transmission Type')
plt.xlabel('Transmission Type')
plt.ylabel('Price (AUD)')
plt.show()
```



Critical Issue 2: Feature Scaling

Problem Identification

In the notebook, the features were encoded but not scaled before putting them into models which can cause issues with models like MLP Regressor, which is sensitive regarding feature scaling.

Without proper scaling, these features can mess up the learning process of the model and lead to poor model performance.

Encoding

We encode the categorical features to integers with OrdinalEncoder and drop the original column for simplicity.

```
: ord_enc = OrdinalEncoder(dtype=int)

data["BrandCode"] = ord_enc.fit_transform(data[["Brand"]])
data["UsedCode"] = ord_enc.fit_transform(data[["UsedOrNew"]])
data["TransmissionCode"] = ord_enc.fit_transform(data[["Transmission"]])
data["DriveTypeCode"] = ord_enc.fit_transform(data[["DriveType"]])
data["FuelTypeCode"] = ord_enc.fit_transform(data[["FuelType"]])
data["LocationCode"] = ord_enc.fit_transform(data[["Location"]])
data["BodyTypeCode"] = ord_enc.fit_transform(data[["BodyType"]])

: # Drop the categorical columns
clean = data.drop(
    columns=['Brand', 'UsedOrNew', 'Transmission', 'DriveType',
            'FuelType', 'Location', 'BodyType'])
```

Now this `clean` will be used for following tasks. We check the shape and statistic info.

```
: print(clean.shape)
clean.describe(include="all")

(14197, 15)
```

Justification

It is important to scale the features before feeding into the models because if the features are not scaled similarly, it can slow down the learning process the model and hinder it from finding solutions efficiently. Scaling all features correctly will help the model interpret features consistently resulting in more accurate outcomes.

Correction

To resolve this issue, use StandardScaler to standardize all the features after encoding them. This modification is particularly important in data preparation for models such as MLP because it maintains uniformity in how features are interpreted.

Encoding

We encode the categorical features to integers with OrdinalEncoder and drop the original column for simplicity.

```
4]: ord_enc = OrdinalEncoder(dtype=int)

data["BrandCode"] = ord_enc.fit_transform(data[["Brand"]])
data["UsedCode"] = ord_enc.fit_transform(data[["UsedOrNew"]])
data["TransmissionCode"] = ord_enc.fit_transform(data[["Transmission"]])
data["DriveTypeCode"] = ord_enc.fit_transform(data[["DriveType"]])
data["FuelTypeCode"] = ord_enc.fit_transform(data[["FuelType"]])
data["LocationCode"] = ord_enc.fit_transform(data[["Location"]])
data["BodyTypeCode"] = ord_enc.fit_transform(data[["BodyType"]])
```

```
5]: # Drop the categorical columns
clean = data.drop(
    columns=['Brand', 'UsedOrNew', 'Transmission', 'DriveType',
            'FuelType', 'Location', 'BodyType'])
```

```
3]: from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
clean = pd.DataFrame(scaler.fit_transform(clean), columns=clean.columns)
```

Now this `clean` will be used for following tasks. We check the shape and statistic info.

```
3]: print(clean.shape)
clean.describe(include="all")
```

(14197, 15)

```
3]:
```

	Year	FuelConsumption	Kilometres	CylindersinEngine	Doors	Seats	
count	1.419700e+04	1.419700e+04	1.419700e+04	1.419700e+04	1.419700e+04	1.419700e+04	1.419700e+04

Critical Issue 3: Poor Feature Selection Method

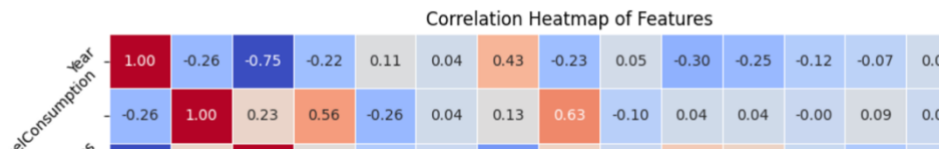
Problem Identification

The notebook selected features by solely picking the top five features with the strongest correlation values to price. While correlation is an important indicator, it might miss out on some non-linear connections that could exist between the features and the target variable. This approach could potentially restrict the models capability in capturing the complexities of the dataset.

Feature Selection & Data Splitting

We first study the correlation between the price and other features.

```
5]: plt.figure(figsize=(14, 10))
sns.heatmap(clean.corr(), annot=True, fmt=".2f", cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap of Features')
plt.xticks(rotation=45)
plt.yticks(rotation=45)
plt.show()
```



Then we keep the 5-top most correlated features and split the dataset. We want the training set the size of 80% of full dataset.

```
] X = clean[['Kilometres', 'Year', 'Displacement', 'CylindersinEngine', 'FuelConsumption']]
y = clean['Price']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=student_id)
```

Justification

Depending on just the correlation alone for choosing features is not enough and can lead to missing out on non-linear relationships to the target variable. Using a better feature selection technique like Recursive Feature Elimination can ensure that the model captures a more complex patterns within the dataset.

Correction

To improve feature selection, use the Recursive Feature Elimination (RFE) technique. RFE uses an optimized feature elimination technique and prioritizes the most important ones for predictive tasks. This method goes beyond linear relationships, to accurately pinpoint the most important features to consider for analysis and prediction purposes.


```
[41]: from sklearn.feature_selection import RFE
      from sklearn.linear_model import LinearRegression

      estimator = LinearRegression()
      rfe = RFE(estimator, n_features_to_select=5)

      X_rfe = rfe.fit_transform(clean.drop(columns=['Price']), clean['Price'])
      selected_features = clean.drop(columns=['Price']).columns[rfe.support_]

      print("Selected Features:", selected_features)

      Selected Features: Index(['Year', 'Kilometres', 'CylindersinEngine', 'DriveTypeCode',
                              'FuelTypeCode'],
                              dtype='object')
```

Based from the correlation matrix and the RFE method, the results showed 3 similar features namely: Year, Kilometers, and Cylinders in Engine. On the other hand the other two features were different for each method. In this case, we are using the five selected features from RFE method because it a more accurate measure of the predictive power of the features, for it handles even the non-linear relationships. Hence, we keep the selected features and split the dataset. We want the training set the size of 80% of full dataset.

```
[43]: X = clean[['Year', 'Kilometres', 'CylindersinEngine', 'DriveTypeCode',
                'FuelTypeCode']]
      y = clean['Price']

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=student_id)
```

Critical Issue 4: Incorrect and Incomplete Analysis of Results

Problem Identification

The initial analysis presented in the report indicates that the Decision Tree Regressor (DTF) is the best option among the predictive models used including Linear Regression (LR), and Multi-Layer Perceptron (MLC). However, the analysis did not explain the evaluation process in terms of the model performance metrics and model suitability.

Analysis

According to the result we have, among these three models (LR, DTR and MLP) the best option for this dataset is DTR. However, the relationships among the features within this dataset are not obvious for these models to catch, thus all the performance are not very satisfying. We might need to further clean the data (e.g. remove outliers) or deploy deep learning models for the prediction.



Justification

When evaluating a models performance accuracy, we need to consider R-squared (R^2), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values for they help in

understanding the prediction capabilities of the models. The R squared values for all the models are quite low which shows that each model is struggling to account for the variations in the target variable. While the high RMSE values visible across all models indicate a significant difference between predicted and actual prices.

Correction

To revise the initial analysis, we can further evaluate the models by analyzing their performance on each performance metrics like Mean Squared Error (MSE) Root Mean Squared Error (RMSE) and R-squared (R^2) values. This is essential in gaining a more comprehensive understanding of the predictive accuracy of the models, how close predicted values are from their actual values, and identifying the factors that may have affected the performance of the models.

 **Analysis** 

After analyzing the performance of three models namely Linear Regression (LR), Decision Tree Regressor (DTR), and Multi Layer Perceptron (MLPR), it was found that none of them were able to predict car prices with high precision. All of their R-squared scores were moderately low (approximately 0.50 for LR, 0.40 for DTR, and 0.48 for MLP). These scores suggest that the models explain a portion of the variability seen in car prices. On the other hand, the Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values in various models furthered the evidence that there is a notable discrepancy between the predicted prices and the actual prices. Although the Linear Regression model demonstrates better performance in terms of R squared (R^2), it may struggle to capture complex, non-linear connections within the data. The Decision Tree Regression model implies that it could potentially be underfitting, which limits its ability to capture intricate patterns effectively. The MLP model is supposed to work with linear connections but because of scaling problems or needing more adjustments, it does not perform as expected. To make the models work better we can consider removing outliers, adjusting features, further tuning of the models, or even exploring other advanced models.