Principles of Data Science Assessment

Name: Farrukh Nizam Arain

1. Data Understanding and Exploration:

1.1 Meaning and Type of Features:

After loading the dataset, the code to know the basic structure of the dataset was ran to know the size and shape of the dataset:

```
# Checking for size and shape of the dataset
print("Size of the Dataset is: ", df.size)
print("Shape of the Dataset is: ", df.shape)
```

```
Size of the Dataset is:  4824060
Shape of the Dataset is:  (402005, 12)
```

To distinguish the type of features, they were segregated into a couple of lists of categorical and numerical columns to know the nature of the features:

```
# Extracting a list of Categorical and Numerical Columns from the dataset
cat_col = df.select_dtypes(include=['object']).columns.tolist()
num_col = df.select_dtypes(exclude=['object']).columns.tolist()

# Printing the list of Categorical and Numerical Columns
print("Categorical Columns:", cat_col)
print("Numerical Columns:", num_col)
```

```
Categorical Columns: ['reg_code', 'standard_colour', 'standard_make',
'standard_model', 'vehicle_condition', 'body_type', 'fuel_type']
Numerical Columns: ['public_reference', 'mileage', 'year_of_registration',
'price', 'crossover_car_and_van']
```

Among the significant number of features to analyse and answer the stakeholder's query, three features are discussed here which were also considered extremely significant during the course of the project:

a) Mileage (mileage):
Mileage is a numerical column in the dataset, and it has a datatype is float64. This feature measures the number of miles that the car has travelled in the past.

b) Year of Registration (year_of_registration):
This is another numerical column, and it has a datatype is also float64. This feature conveys the year on which the car was registered in the past.

c) Manufacturer's Name (standard_make):
It is a categorical column, and it has an object data type. This feature conveys the name of the manufacturer that built a vehicle.

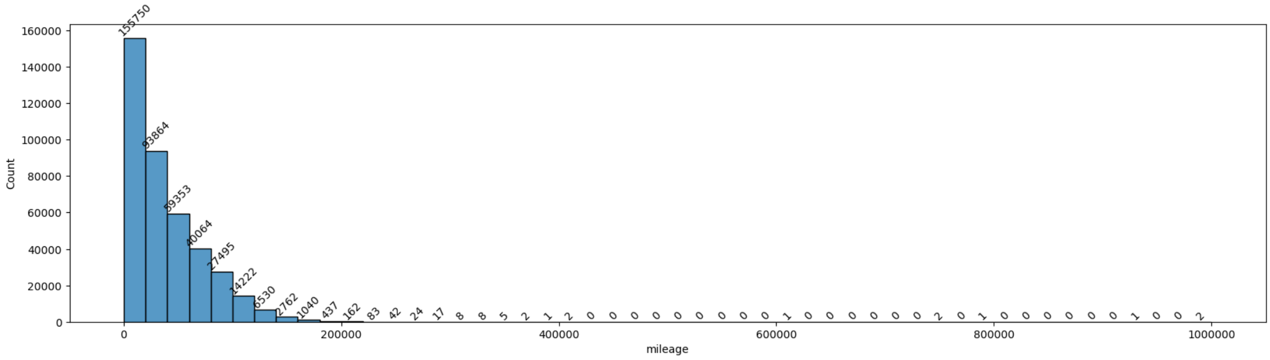
d) Colour of the vehicle (standard_colour):
This is also a categorical column, and it has an object data type. This feature stores the original colour on the body of the vehicle.

1.2 Analysis of Distributions:

The entire dataset was analysed extensively before the initialization of data pre-processing stage, but there are quite a few interesting revelations that should be mentioned below:

a) Mileage (mileage):

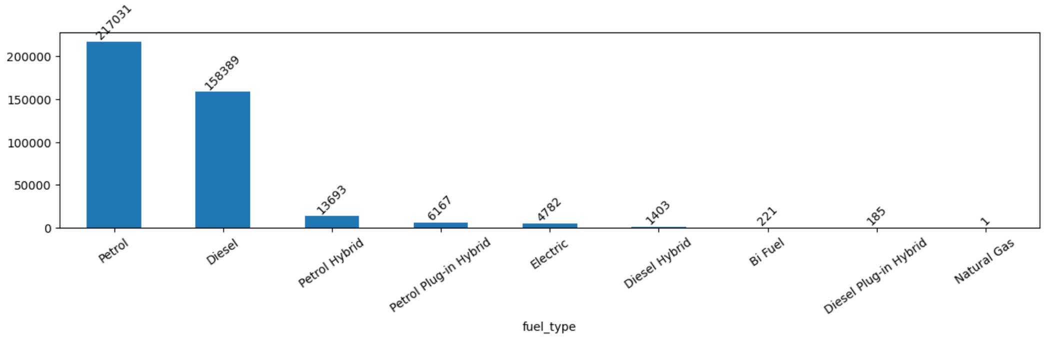
```
# Code snippet for generating a histogram for mileage feature
plt.figure(figsize=(20, 5))
plt.ticklabel_format(style='plain')
ax = sns.histplot(df['mileage'], bins=50, binrange=(0, 1000000))
# Labelling bars in the histogram
for i in ax.containers:
    ax.bar_label(i, rotation= 45, label_type="edge")
plt.show()
```



This histogram revealed interesting facts about mileage of the cars. Firstly, majority of the vehicles' mileage were in between the range of 0 and 180,000. This mean that most cars are either entirely new or slightly old based on the data.

b) Fuel Type (fuel_type):

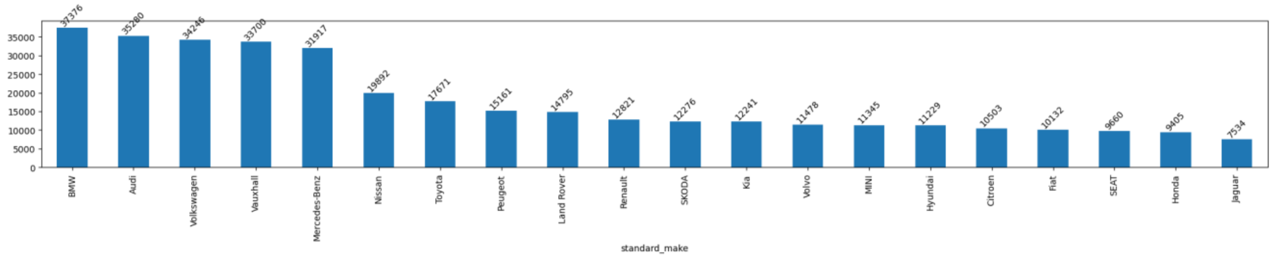
```
# Code snippet for generating a bar graph for fuel_type feature
plt.figure(figsize=(15, 3))
plt.ticklabel_format(style='plain')
ax = df['fuel_type'].value_counts().plot(kind='bar')
for i in ax.containers:
    ax.bar_label(i, rotation= 45, label_type="edge")
plt.show()
```



Bar plot revealed that majority of the vehicles have either petrol or diesel fuel systems embedded. These systems are associated with medium or lower budgeted vehicles. On the contrary, newer fuel systems like petrol hybrid, electric and Bi Fuel are in lesser number. One major reason of them in lesser quantity is that fuel systems of petrol and diesel are most widely used around the world because of being economical in terms of installation, maintenance, and refuelling costs.

c) Manufacturer's Name (standard_make):

```
# Code snippet for generating a bar plot for standard_make feature
plt.figure(figsize=(25, 3))
plt.ticklabel_format(style='plain')
ax = df['standard_make'].value_counts().nlargest(20).plot(kind='bar')
for i in ax.containers:
    ax.bar_label(i, rotation= 45, label_type="edge")
plt.show()
```



Bar plot concerning the value distribution in terms of manufacturers revealed that most BMW, Volkswagen, Nissan, Toyota, and similar brands are prevalent in numbers which are known to generally produce cheaper cars which are known to attract customers from different financial classes and budget. Additionally, this single plot also revealed that most of the price points are on the cheaper side.

2. Data Pre-Processing:

2.1 Data Cleaning (e.g., dealing with incorrect values, outliers) (2-3)

Several activities were performed for pre-processing the data before the analysis or visualization stage, but there were a few which revealed some key insights.

a) Outliers:

```
# Detecting any possible outliers in the year_of_registration column
yreg_counts =
df['year_of_registration'].value_counts().to_frame().reset_index().sort_values
(by='year_of_registration', ascending=True)
yreg_counts.columns = ['year_of_registration', 'count']
pd.set_option('display.max_rows', None)
print(yreg_counts)
```

```
    year_of_registration  count
68                 999.0      3
74                1006.0      1
63                1007.0      3
73                1008.0      1
69                1009.0      2
...
```

The values which were less than and equal to 1909.0 were removed because either the car launch's year was after the year of registration value or automobiles weren't invented by then.

```
# Dropping the rows of the column having values less than 1909.0
df.drop(df[df['year_of_registration'] <= 1909.0].index, inplace=True)
```

b) Missing values:

```
# Checking for missing values in the dataset
df['standard_colour'].isnull().sum()
```

```
5378
```

```
# Taking a mode of standard_colour for each standard_make and standard_model
combination
standard_model_mode_series = df.groupby(['standard_make',
'standard_model'])['standard_colour'].transform(lambda x: x.mode().iloc[0] if
not x.mode().empty else pd.NA)
df['standard_colour'] =
df['standard_colour'].fillna(standard_model_mode_series)
```

For the column pertaining to body colour of the vehicles, columns related to vehicle's make and model were grouped together and value of the mode was extracted and replaced in the relevant locations having NaN values. Similarly, missing values in the other features were mostly handled by taking mean for the numerical values and mode for the non-numerical ones.

After cleaning of all of the values via statistical methods, the rest of the missing values were deleted straightaway as no combination or method was available at that time to retain the rest of the 155 rows.

```
# Deleting all the remaining missing values in the dataset
df.dropna(inplace=True)
```

```
#Rechecking whether all the missing values have been handled
df.isna().sum()
```

```
public_reference         0
mileage                  0
reg_code                 0
standard_colour          0
standard_make            0
standard_model           0
vehicle_condition        0
year_of_registration     0
price                    0
body_type                0
crossover_car_and_van    0
fuel_type                0
dtype: int64
```

```
# Checking out the shape of the dataset after handling missing values
df.shape
```

```
(401872, 12)
```

Out of the 402,005 observations in the original dataset, 99.97 percent of the values were cleaned based on the existing related values, and because of that 401,872 observations were ready for the next phase of analysis.

2.2 Feature Engineering:

There were a few critical decisions that were taken regarding modelling of the dataset:

a) Separating the statuses of new and used vehicles:

```
# According to assumption, mileage up to 100 miles is considered as new car,
so we will replace status with `NEW` for mileage less than or equals to 100
miles
mask = (df['vehicle_condition'] == 'USED') & (df['mileage'] <= 100)
x = df[mask].sort_values(by='mileage',
ascending=False)['vehicle_condition'].replace('USED', 'NEW')
df.update(x)
```

There were a number of vehicles in the feature for vehicle’s condition which had both NEW and USED statuses, and also mileage of 100 miles. So, to segregate both, status of vehicles having miles equal, or less than one hundred miles were changed from USED to NEW. Furthermore, the data frame was queried to confirm there is no overlapping in between both the statuses.

```
# Checking whether the vehicle_condition has been updated
df.query('vehicle_condition == "USED" and mileage <= 100')
```

b) New column for age of the vehicle:

```
# Creating a new column for age of the vehicle
df['age_of_vehicle'] = 2021 - df['year_of_registration']
```

As the dataset is four years old and data regarding to inflation rate, depreciation rate, market demand & trends, regional price differences, and model-specific factors was not available so it was assumed that the age of the vehicle would be calculated and assumed by subtracting 2021 from year of registration for age calculation.

c) New column for price categories:

```
# Declaring price bins and it's related labels for the used cars
price_bins = [0, 13999, 17999, 21999, 25999, 29999, 33999, 37999,
float('inf')]
price_labels = ['Very Low', 'Low', 'Medium-Low', 'Medium', 'Medium-High',
'High', 'Very High', 'Luxury']

# Executing the query to select used car condition & inseting the price
category in the dataset
df['price_category'] = pd.cut(df.query('vehicle_condition ==
"USED"')['price'], bins=price_bins, labels=price_labels, right=True)
```

```
# Declaring price bins and it's related labels for new cars
price_bins = [0, 11999, 16999, 21999, 35999, 49999, 69999, 99999,
float('inf')]
price_labels = ['Very Low', 'Low', 'Medium-Low', 'Medium', 'Medium-High',
'High', 'Very High', 'Luxury']

df.loc[df['vehicle_condition'] == 'NEW', 'price_category'] =
pd.cut(df.loc[df['vehicle_condition'] == "NEW", 'price'], bins=price_bins,
labels=price_labels, right=True)
```

```
# Checking if there are NaN values in the dataset
df.isna().sum()
```

```
public_reference         0
mileage                  0
reg_code                 0
standard_colour          0
standard_make            0
standard_model           0
vehicle_condition        0
year_of_registration     0
price                    0
body_type                0
crossover_car_and_van    0
fuel_type                0
age_of_vehicle           0
price_category           0
dtype: int64
```

A new column for price categories was created using a new function pd.cut() after segregating and labelling the price values into almost equal-sized quantiles / bins. All those categories were assumed after getting the price ranges from Honest John for realistic categories for analysis and visualization. Furthermore, similar code was executed for cars where vehicle condition was inputted as NEW, and data was researched and assumed from different websites like Motors.co.uk.

2.3. Subsetting

This phase of the assessment pertained to selection of related features and rows:

a) Feature Selection:
All the features were considered for analysis except for public reference and registration code because it did not have any direct association with price. Some of the key features were:

i) Manufacturer's Name (standard_make):
Among all the categorical features in the dataset, this feature was considered incredibly significant in having impact on price. Although it may seem that the manufacturer's name does not have an impact on price, but the reputation associated with a manufacturer's name does have an influence on not only the price of the vehicle, but it also helps in retaining a vehicle's value for a longer period.

ii) Mileage (mileage):
Mileage is one of the numerical features which was regarded as one of the most critical in successful accomplishment of the project, even during the early stages of the project because generally prices are highly dependent on this value. If an automobile has a high mileage, then it is considered that it has a significant deterioration since it was registered after being bought.

b) Row Sampling:
Although whole of the dataset was used in the earlier processes, a small sample or subset was utilized for the last section of analysis and recommendation via visualization:

* To reduce computational time and cognitive load.
* To view and interpret the trends more clearly and precisely.
* To remove bias and generating a representative sample, a random fraction of the dataset for 'every year' will be extracted so that both old and new cars, irrespective of when they were registered and how many data points they have, will be included so it will have equal representation without elimination of specific years or model of a car.

```
# Taking a sample from the dataset for this visualization and name it as df1
df1 = df.groupby('year_of_registration').sample(frac=0.01, random_state=1)

# Extracting the number of values as per the year of registration
df1['year_of_registration'].value_counts()
```

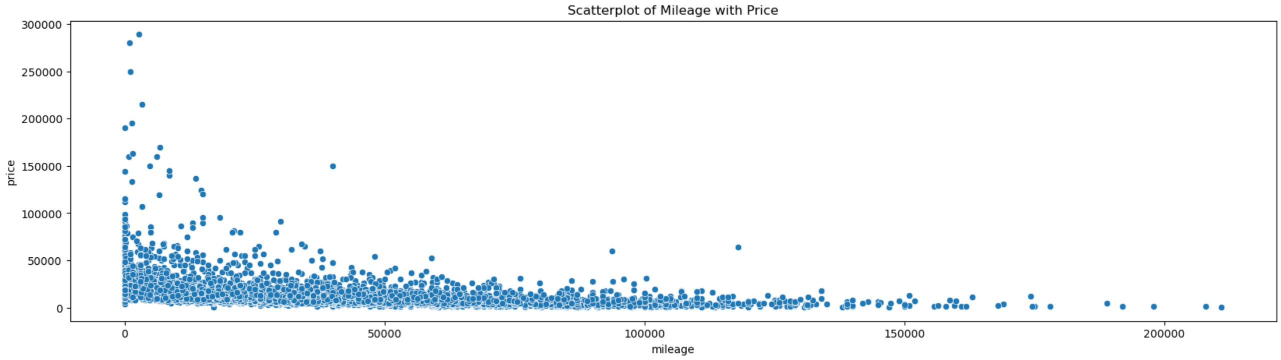
```
year_of_registration
2017    850
2016    480
2019    433
2018    397
2020    349
2015    291
...
Name: count, dtype: int64
```

3. Analysis of Associations and Group Differences:

3.1. Quantitative-Quantitative:

a) Association of Mileage with Price:

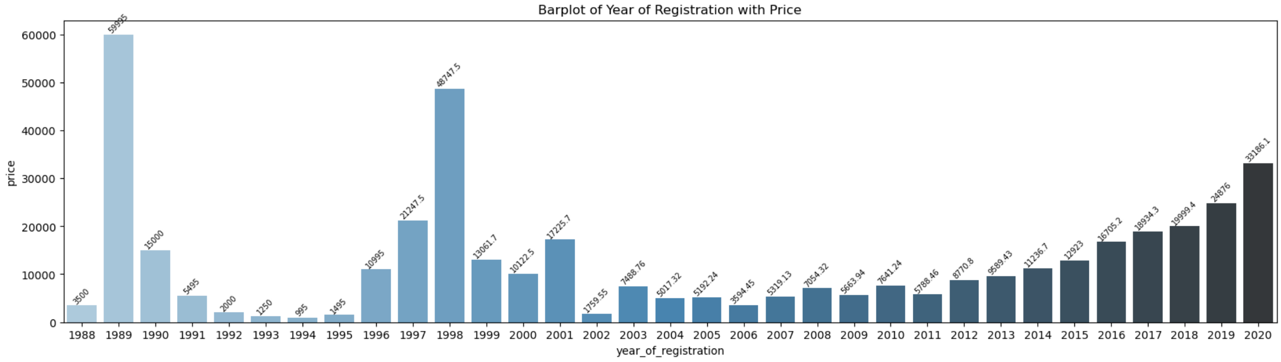
```
# Scatterplot of mileage with price
plt.figure(figsize=(20, 5))
plt.ticklabel_format(style='plain')
sns.scatterplot(data=df1, x='mileage', y='price')
plt.title("Scatterplot of Mileage with Price")
plt.show()
```



According to the first scatterplot, whenever mileage increased, then price of the car decreased. So, there was a negative but strong association in between both.

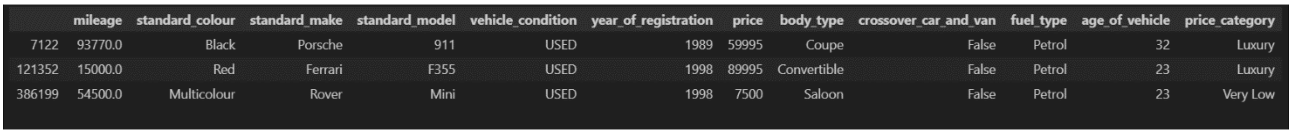
b) Association of Year of Registration with Price:

```
# Bar plot of year of registration with price
plt.figure(figsize=(20, 5))
plt.ticklabel_format(style='plain')
ax = sns.barplot(data=df1, x='year_of_registration', y='price', errorbar=None,
palette='Blues_d')
for i in ax.containers:
    ax.bar_label(i,label_type="edge", size=7, rotation= 45)
plt.title("Barplot of Year of Registration with Price")
plt.show()
```



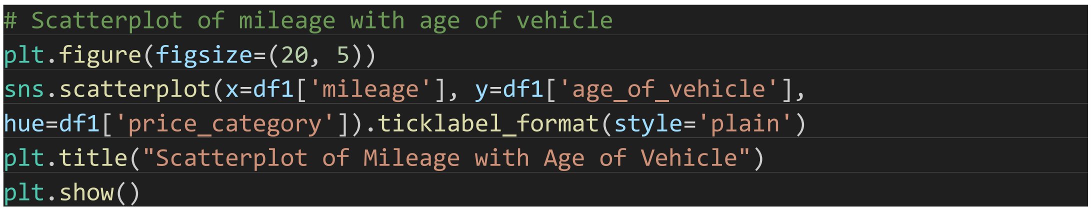
Overall, there was a positive and strong association in between year of registration and price of the car in the second bar plot. However, there were a couple irregular tall bars in 1998 and 1989 and it was because of the two luxury cars which are up for sale. Overall, it was a clear trend that recently registered cars have higher prices if you compare them with those which were registered further in the past.

```
df1.query('year_of_registration == 1989 | year_of_registration == 1998 ')
```

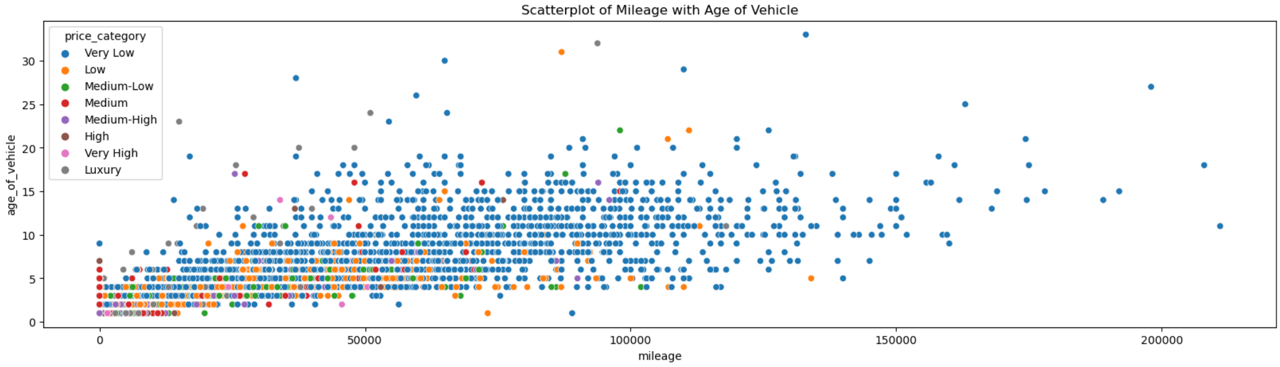


	mileage	standard_colour	standard_make	standard_model	vehicle_condition	year_of_registration	price	body_type	crossover_car_and_van	fuel_type	age_of_vehicle	price_category
7122	93770.0	Black	Porsche	911	USED	1989	59995	Coupe	False	Petrol	32	Luxury
121352	15000.0	Red	Ferrari	F355	USED	1998	89995	Convertible	False	Petrol	23	Luxury
386199	54500.0	Multicolour	Rover	Mini	USED	1998	7500	Saloon	False	Petrol	23	Very Low

c) Association of Mileage and Age of Vehicle:



```
# Scatterplot of mileage with age of vehicle
plt.figure(figsize=(20, 5))
sns.scatterplot(x=df1['mileage'], y=df1['age_of_vehicle'],
hue=df1['price_category']).ticklabel_format(style='plain')
plt.title("Scatterplot of Mileage with Age of Vehicle")
plt.show()
```



* In the third scatterplot which is in between mileage and age of the vehicle, it was observed that price decreased with the increase in mileage and age of vehicle for all the cars. This is due to the deteriorating condition and value of the vehicle in time and usage. Moreover, colour coding the data points based on the price category established this hypothesis.

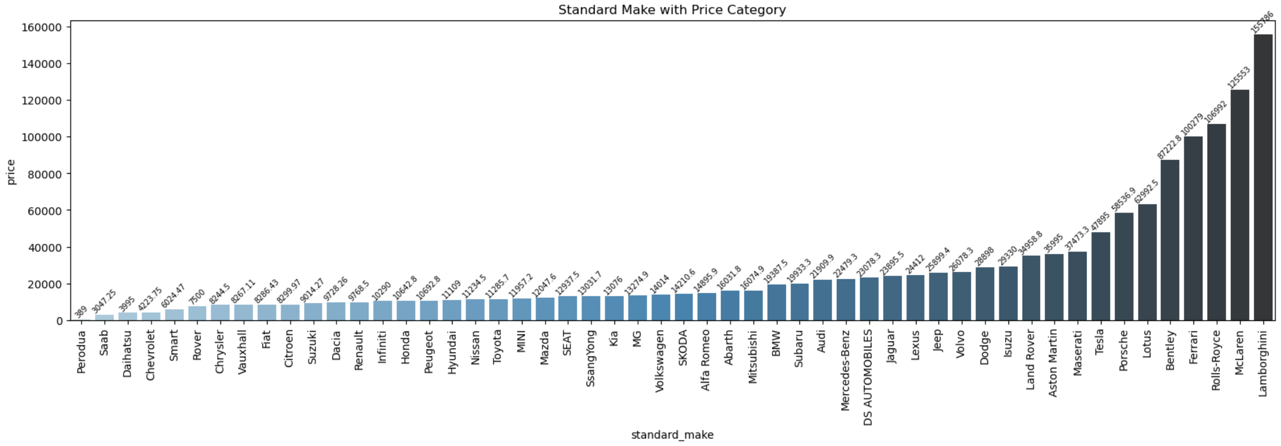
Recommendation(s):

After analysis of the patterns, it is analysed that customers with newer & cheaper vehicles with lower mileage are in large numbers. So, it is suggested that AutoTrader can introduce some incentives, like free registration for such customers to further increase the number of car sales and can earn higher profit for them.

3.2. Quantitative-Categorical:

a) Association of Name of Manufacturer with Price:

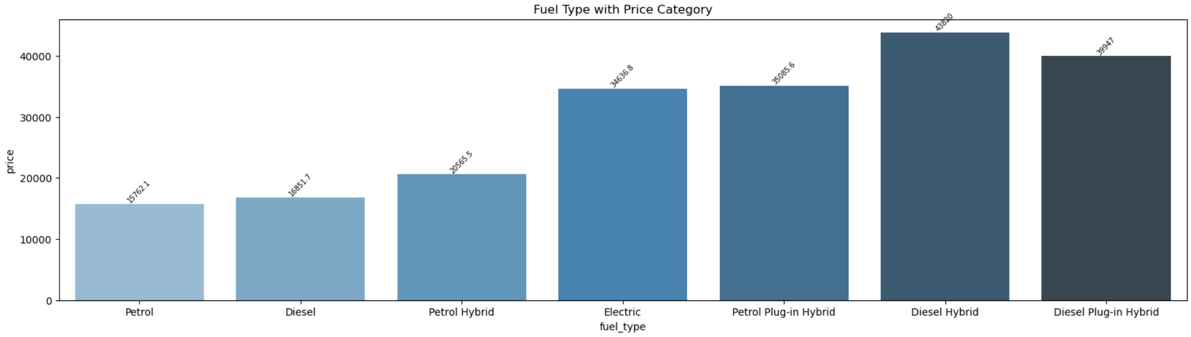
```
# Bar plot of standard_make with price
plt.figure(figsize=(20, 5))
plt.ticklabel_format(style='plain')
ax =
sns.barplot(data=df1.groupby('standard_make')['price'].mean().reset_index().so
rt_values(by='price', ascending=True), x='standard_make', y='price',
errorbar=None, palette='Blues_d')
plt.xticks(rotation=90)
for i in ax.containers:
    ax.bar_label(i, rotation=45 ,label_type="edge", size=7)
plt.title("Standard Make with Price Category")
plt.show()
```



When standard make was compared with price in the first plot, it was quite evident that the some specific manufacturers like Lamborghini, McLaren, Rolls-Royce were associated with higher prices of cars than other manufacturers like Toyota, Nissan, etc. That is why they were priced higher & it reflected that manufacturer's image / reputation played a key role in price.

b) Association of Fuel Type with Price:

```
# Bar plot of fuel type with price
plt.figure(figsize=(20, 5))
plt.ticklabel_format(style='plain')
ax = sns.barplot(data=df1.sort_values('price', ascending=True), x='fuel_type',
y='price', errorbar=None, palette='Blues_d')
for i in ax.containers:
    ax.bar_label(i, rotation=45 ,label_type="edge", size=7)
plt.title("Fuel Type with Price Category")
plt.show()
```



In the second plot, fuel type column of a car had significant impact on the price. For instance, petrol and diesel cars are more widely acceptable, cheaper, and have access to a wider number of populations. On the contrary, even though newer technologies of hybrid electric cars are efficient, but cost of manufacturing, buying, and refuelling options for the customers are expensive. That is why cars having conventional fuel systems like petrol & diesel are priced lower.

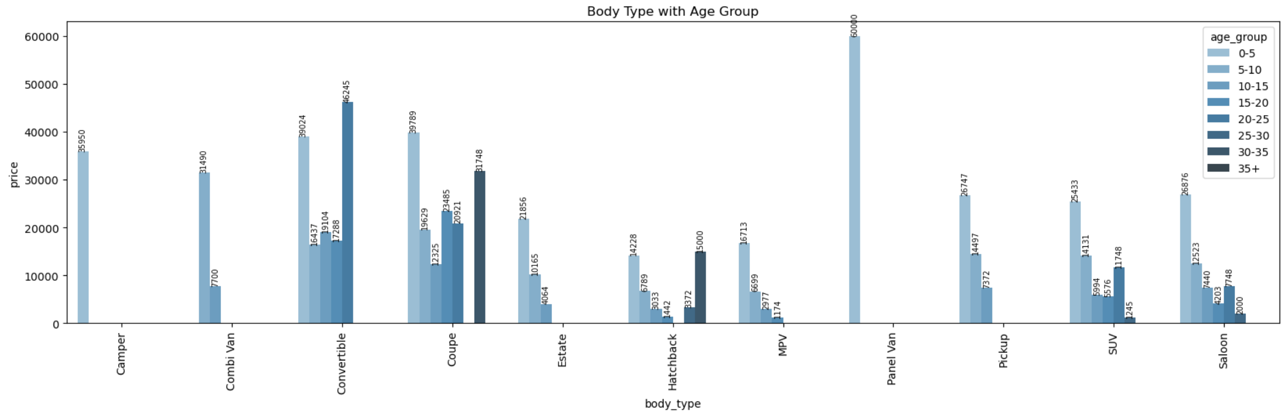
c) Association of Body Type with Price:

```
# Declaring the bins and it's related labels
v_age_bins = [0, 5, 10, 15, 20, 25, 30, 35, float('inf')]
v_age_labels = ['0-5', '5-10', '10-15', '15-20', '20-25', '25-30', '30-35',
'35+']

# Copying the dataset to a new dataframe
df_group = df1.copy()
df_group['age_group'] = pd.cut(df_group['age_of_vehicle'], bins=v_age_bins,
labels=v_age_labels, right=True)

grouped_dataset = df_group.groupby(['body_type',
'age_group'])['price'].mean().round().reset_index()

# visualizing the grouped dataset
plt.figure(figsize=(20, 5))
plt.ticklabel_format(style='plain')
ax = sns.barplot(data=grouped_dataset, x='body_type', y='price',
hue='age_group', palette='Blues_d')
plt.xticks(rotation=90)
for i in ax.containers:
    ax.bar_label(i, rotation=90 ,label_type="edge", size=7)
plt.title("Body Type with Age Group")
plt.show()
```



Grouped bar chart gave an estimate as to how much age of a vehicle impacted the body type of all the vehicles available in the sample, irrespective of the brand. Hatchback was the worst when it came to retaining its value & effects as far as age of vehicle was concerned. On the contrary, convertible, and then coupe were the types which retained its price much better than others.

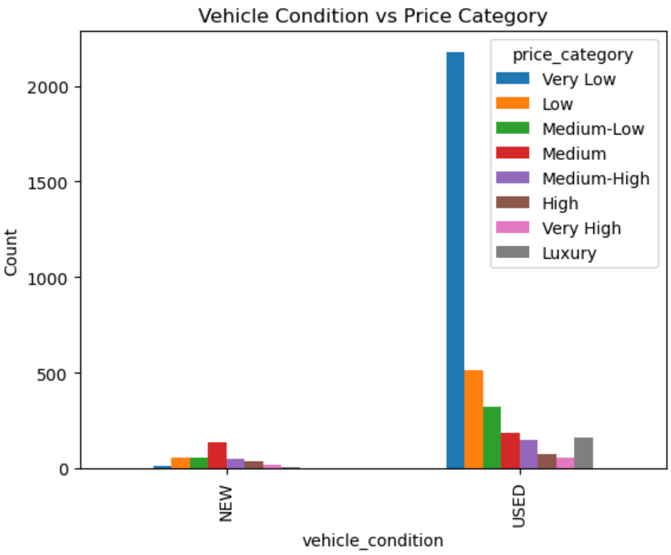
Recommendation(s):

Most of the existing cars have conventional petrol and diesel fuel systems fitted in them. So, relevant stakeholders at the company can further emphasize and target the customers with similar needs to boost sales and profits. In addition to that, as convertible and coupe body types retain its values much longer than other ones, marketing team can increase the number of new or old vehicles with same body type to convince the prospective customers that the purchase of such vehicles can help you retain the value of the cars to a much longer period.

3.3. Categorical-Categorical:

a) Association of Vehicle Condition with Price Category:

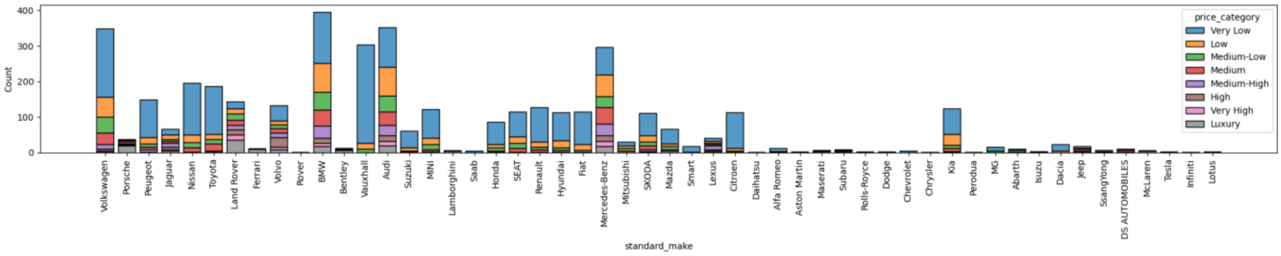
```
# stacked bar chart of vehicle_condition with price category
plt.figure(figsize=(20, 5))
pd.crosstab(df1['vehicle_condition'], df1['price_category']).plot(kind='bar',
stacked=False)
plt.ylabel('Count')
plt.title('Vehicle Condition vs Price Category')
plt.show()
```



As per the standards by AutoTrader when it came to classification of vehicle's condition, it did have a strong association with price as there were far more choices in terms of budget for a customer who is interested in buying a vehicle under the tag of USED.

b) Association of Manufacturer's Name with Price Category:

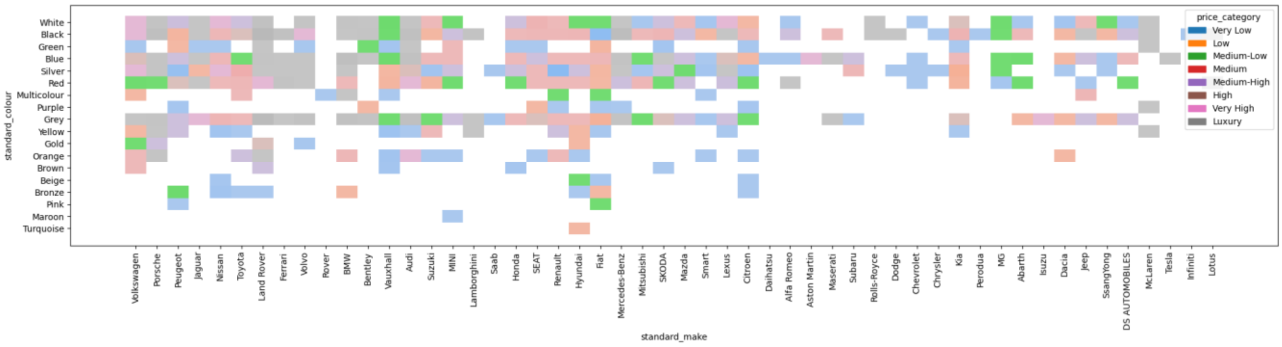
```
# stacked bar chart of standard_make with price category
plt.figure(figsize=(25, 3))
sns.histplot(data=df1, x='standard_make', hue='price_category', multi-
ple='stack', shrink=0.8)
plt.xticks(rotation=90)
plt.show()
```



When it came to the second bar plot of having vehicle's make on x axis, budget brands such as Volkswagen, Nissan, Toyota, Vauxhall had a higher influence to offer a wide range on different prices, especially in the `Very Low`, 'Low', and 'Medium' price ranges.

c) Association of Car's Model with its Colours:

```
# Histplot in between car model and condition with hue as price category
plt.figure(figsize=(25, 5))
sns.histplot(data=df1, x='standard_make', y='standard_colour',
hue='price_category', multiple='stack', legend=True)
plt.xticks(rotation=90)
plt.show()
```



In the third plot, different vehicle's make and colour combinations showed its association with a different price category in the dataset. For instance, Land Rover's specific colours like Black and White are priced higher than Bronze, which is situated at the 'Very Low' category. So, all brands have cars based on different price categories according to the colour that they sell which is popular in the market.

Recommendation(s):

Relevant stakeholders can market the vehicle's variable price points already in the database in such a way that customer with every budget can buy and sell their dream cars through them at reasonable rates and at a quicker rate. Similarly, it will also attract customers who want to sell their cars because they will know that the chances of getting a customer will be higher if they put their cars up for sale at AutoTrader. Moreover, there are cars of different price categories of some brands which are known to produce cheaper and durable cars like Volkswagen, Nissan, Toyota, BMW, and other to. AutoTrader can have a partnership with them to convince that they have a significant customer base and offer them discounts if they purchase new cars of the same brand.