What drives the price of a car?

OVERVIEW

In this application, you will explore a dataset from kaggle. The original dataset contained information on 3 million used cars. The provided dataset contains information on 426K cars to ensure speed of processing. Your goal is to understand what factors make a car more or less expensive. As a result of your analysis, you should provide clear recommendations to your client -- a used car dealership -- as to what consumers value in a used car.

CRISP-DM Framework

To frame the task, throughout our practical applications we will refer back to a standard process in industry for data projects called CRISP-DM. This process provides a framework for working through a data problem. Your first step in this application will be to read through a brief overview of CRISP-DM [here (https://mo-pcco.s3.us-east-1.amazonaws.com/BH-PCMLAI/module_11/readings_starter.zip)](https://mo-pcco.s3.us-east-1.amazonaws.com/BH-PCMLAI/module_11/readings_starter.zip). After reading the overview, answer the questions below.

Business Understanding

From a business perspective, we are tasked with identifying key drivers for used car prices. In the CRISP-DM overview, we are asked to convert this business framing to a data problem definition. Using a few sentences, reframe the task as a data task with the appropriate technical vocabulary.

The goal of this project is to find out the factors which affect a car's value, this can be done using tabular data with corresponding features listed out. Since multiple quantifiable factors are correlated with a car's value over time, all of them can be used to predict a certain car's value with relatively good accuracy, considering the fact that the data is plentiful and the correlation between the individual features and the car's value is high.

Data Understanding

After considering the business understanding, we want to get familiar with our data. Write down some steps that you would take to get to know the dataset and identify any quality issues within. Take time to get to know the dataset and explore what information it contains and how this could be used to inform your business understanding.

In [57]:
```
import pandas as pd
import numpy as np
```

In [58]:
```
df=pd.read_csv('vehicles.csv')
print(df.tail())
print(df.shape)
```

```
                id   region  price    year manufacturer  \
426875  7301591192  wyoming  23590  2019.0       nissan   
426876  7301591187  wyoming  30590  2020.0        volvo   
426877  7301591147  wyoming  34990  2020.0     cadillac   
426878  7301591140  wyoming  28990  2018.0        lexus   
426879  7301591129  wyoming  30590  2019.0          bmw   

                           model condition    cylinders    fuel  odometer  
\
426875         maxima s sedan 4d      good  6 cylinders     gas   32226.0   
426876  s60 t5 momentum sedan 4d      good          NaN     gas   12029.0   
426877          xt4 sport suv 4d      good          NaN  diesel    4174.0   
426878           es 350 sedan 4d      good  6 cylinders     gas   30112.0   
426879  4 series 430i gran coupe      good          NaN     gas   22716.0   

       title_status transmission                VIN drive size       type  
\
426875        clean        other  1N4AA6AV6KC367801   fwd  NaN      sedan   
426876        clean        other  7JR102FKXLG042696   fwd  NaN      sedan   
426877        clean        other  1GYFZFR46LF088296   NaN  NaN  hatchback   
426878        clean        other  58ABK1GG4JU103853   fwd  NaN      sedan   
426879        clean        other  WBA4J1C58KBM14708   rwd  NaN      coupe   

       paint_color state  
426875         NaN    wy  
426876         red    wy  
426877       white    wy  
426878      silver    wy  
426879         NaN    wy  
(426880, 18)
```

1. Identifying gaps in the data, such as Nan values

In [59]:
```
for i in df.columns:
    print(f"{i}:{df[i].isnull().values.sum()}")
```

```
id:0
region:0
price:0
year:1205
manufacturer:17646
model:5277
condition:174104
cylinders:177678
fuel:3013
odometer:4400
title_status:8242
transmission:2556
VIN:161042
drive:130567
size:306361
type:92858
paint_color:130203
state:0
```

2. Analyzing the mean, variance, standard deviation and other aspects of the data

In [60]:
```
df.describe()
```

Out[60]:

	id	price	year	odometer
count	4.268800e+05	4.268800e+05	425675.000000	4.224800e+05
mean	7.311487e+09	7.519903e+04	2011.235191	9.804333e+04
std	4.473170e+06	1.218228e+07	9.452120	2.138815e+05
min	7.207408e+09	0.000000e+00	1900.000000	0.000000e+00
25%	7.308143e+09	5.900000e+03	2008.000000	3.770400e+04
50%	7.312621e+09	1.395000e+04	2013.000000	8.554800e+04
75%	7.315254e+09	2.648575e+04	2017.000000	1.335425e+05
max	7.317101e+09	3.736929e+09	2022.000000	1.000000e+07

3. Identifying the data types of individual features

In [61]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 426880 entries, 0 to 426879
Data columns (total 18 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   id            426880 non-null  int64  
 1   region        426880 non-null  object 
 2   price         426880 non-null  int64  
 3   year          425675 non-null  float64
 4   manufacturer  409234 non-null  object 
 5   model         421603 non-null  object 
 6   condition     252776 non-null  object 
 7   cylinders     249202 non-null  object 
 8   fuel          423867 non-null  object 
 9   odometer      422480 non-null  float64
 10  title_status  418638 non-null  object 
 11  transmission  424324 non-null  object 
 12  VIN           265838 non-null  object 
 13  drive         296313 non-null  object 
 14  size          120519 non-null  object 
 15  type          334022 non-null  object 
 16  paint_color   296677 non-null  object 
 17  state         426880 non-null  object 
dtypes: float64(2), int64(2), object(14)
memory usage: 58.6+ MB
```

4. Identifying the unnecessary features

In [62]:
```
df.columns
```

Out[62]:
```
Index(['id', 'region', 'price', 'year', 'manufacturer', 'model', 'conditi
on',
       'cylinders', 'fuel', 'odometer', 'title_status', 'transmission',
'VIN',
       'drive', 'size', 'type', 'paint_color', 'state'],
      dtype='object')
```

id, VIN and state are unnecessary.

Data Preparation

After our initial exploration and fine tuning of the business understanding, it is time to construct our final dataset prior to modeling. Here, we want to make sure to handle any integrity issues and cleaning, the engineering of new features, any transformations that we believe should happen (scaling, logarithms, normalization, etc.), and general preparation for modeling with `sklearn`.

In [63]:
```
df=df.drop(columns=['id','VIN','state'])
```

In [64]:
```
df=df.dropna()
```

In [65]:
```
num_cols=[]
categ_cols=[]
for i in df.columns:
    if df[i].dtype==np.float64 or df[i].dtype==np.int64:
        num_cols.append(i)
    else:
        categ_cols.append(i)
num_cols.remove('price')
print(categ_cols,num_cols,df.columns)
```

```
['region', 'manufacturer', 'model', 'condition', 'cylinders', 'fuel', 'ti
tle_status', 'transmission', 'drive', 'size', 'type', 'paint_color'] ['ye
ar', 'odometer'] Index(['region', 'price', 'year', 'manufacturer', 'mode
l', 'condition',
       'cylinders', 'fuel', 'odometer', 'title_status', 'transmission',
       'drive', 'size', 'type', 'paint_color'],
      dtype='object')
```

In [66]:
```
from sklearn.compose import make_column_transformer, TransformedTargetRegre
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
```

In [67]:
```
imputer=SimpleImputer(missing_values = np.nan ,strategy='mean')
ct=make_column_transformer(
    (imputer, num_cols),
     (OneHotEncoder(handle_unknown='ignore'), categ_cols),remainder='passth
```

In [68]:
```
X=df.drop(columns=['price'])
y=df[['price']]
```

In [69]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_sta
X_train
```

Out[69]:

	region	year	manufacturer	model	condition	cylinders	fuel	odometer	title_s
98961	jacksonville	2017.0	dodge	charger	excellent	8 cylinders	gas	2439.0	
388009	vermont	1985.0	bmw	318i 2-door coupe	good	4 cylinders	gas	133000.0	
156216	cedar rapids	2016.0	honda	cr-v	good	4 cylinders	gas	95789.0	
150501	fort wayne	2016.0	nissan	sentra sv	like new	4 cylinders	gas	94395.0	
222651	springfield	2003.0	chevrolet	silverado 1500hd	good	8 cylinders	gas	214000.0	
...	...	...	...	...	...	...	...	...	
351208	sioux falls / SE SD	2013.0	chevrolet	malibu lt	excellent	4 cylinders	gas	118000.0	
250467	central NJ	1974.0	chevrolet	monte carlo	excellent	8 cylinders	gas	40000.0	
259603	albuquerque	2002.0	chevrolet	silverado	excellent	8 cylinders	diesel	230000.0	sa
373168	el paso	2004.0	ford	expedition	good	8 cylinders	gas	202000.0	
211729	duluth / superior	2016.0	ford	explorer	like new	6 cylinders	gas	91299.0	

63356 rows × 14 columns

Modeling

With your (almost?) final dataset in hand, it is now time to build some models. Here, you should build a number of different regression models with the price as the target. In building your models, you should explore different parameters and be sure to cross-validate your findings.

In [70]:
```
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV
from sklearn.linear_model import LinearRegression, Ridge, Lasso
```

In [71]:
```
def pipe(regr,custom_transform):
    pipeline=Pipeline([
            ('transformer', custom_transform),
            ('regressor', regr),
        ])
    return pipeline
```

In [72]:
```
regr=LinearRegression()
model1=pipe(regr,ct).fit(X_train,y_train)
```

In [73]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(model1,X_train,y_train,cv=5,scoring='r2').mean()
```

Out[73]:
```
-5296.934262165079
```

In [74]:
```
ridge=Ridge()
parameters={'alpha':[1e-15,1e-10,1e-8,1e-3,1e-2,1,5,10,20,30,35,40,45,50,55
ridge_regressor=GridSearchCV(ridge,parameters,scoring='r2',cv=5)
ridge_pipe=pipe(ridge_regressor,ct)
ridge_pipe.fit(X_train,y_train)
```

Out[74]:
```
Pipeline(steps=[('transformer',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('simpleimputer',
                                                  SimpleImputer(),
                                                  ['year', 'odometer']),
                                                 ('onehotencoder',
                                                  OneHotEncoder(handle_un
known='ignore'),
                                                  ['region', 'manufacture
r',
                                                   'model', 'condition',
                                                   'cylinders', 'fuel',
                                                   'title_status',
                                                   'transmission', 'driv
e',
                                                   'size', 'type',
                                                   'paint_color'])])),
                ('regressor',
                 GridSearchCV(cv=5, estimator=Ridge(),
                              param_grid={'alpha': [1e-15, 1e-10, 1e-08,
0.001,
                                                    0.01, 1, 5, 10, 20, 3
0, 35,
                                                    40, 45, 50, 55, 10
0]},
                              scoring='r2'))])
```

In [75]:
```
cross_val_score(ridge_pipe,X_train,y_train,cv=5,scoring='r2').mean()
```

Out[75]:
```
-2.0715163217809374
```

Evaluation

With some modeling accomplished, we aim to reflect on what we identify as a high quality model and what we are able to learn from this. We should review our business objective and explore how well we can provide meaningful insight on drivers of used car prices. Your goal now is to distill your findings and determine whether the earlier phases need revisitation and adjustment or if you have information of value to bring back to your client.

Based on the information provided, it is apparent that ridge regression performed better, so increasing the complexity of the model alongside testing it with different parameters may help us make an informed decision on whether ridge regression, linear regression or any other modeling technique needs to be implemented. However, if the accuracy of the newer models is relatively low and not providing valuable information to the group of car dealers, then it may be necessary to revisit and adjust the hyperparameters of the models. Additionally, it is important to consider the context of the data provided and how it is being used by the car dealers. If more detailed analysis and insights are needed to better inform their decisions, then it may be worthwhile to explore other methods such as deep learning algorithms.

Deployment

Now that we've settled on our models and findings, it is time to deliver the information to the client. You should organize your work as a basic report that details your primary findings. Keep in mind that your audience is a group of used car dealers interested in fine tuning their inventory.

REPORT

This report is intended to provide an overview of the performance of two models for predicting used car prices: linear regression and ridge regression. Our primary finding is that the use of ridge regression yields better results than linear regression when it comes to predicting used car prices. The r2 score for ridge regression with grid search implemented for hyperparameters gave a much better result than linear regression, indicating that model complexity is a huge factor in the final accuracy of the results.

In order to evaluate the two models, we gathered data on used car prices from public information sources. We used a variety of criteria to define and quantify our variables, including the age of the vehicle, the vehicle manufacturer, the model, condition, cylinders, fuel, transmission. We then proceeded to compare the performance of our two models: linear regression and ridge regression.

The results of our comparison indicated that the use of ridge regression yields a huge improvement over linear regression when it comes to predicting used car prices. This result is consistent with our prior studies comparing ridge regression and linear regression for predicting prices of other items.

We believe that these results can be particularly beneficial for used car dealers. By leveraging the predictive qualities of ridge regression models, used car dealers will be able to anticipate changes in the market and purchase the inventory that is likely to yield the best rewards.

In conclusion, based on our analysis of the performance of linear regression and ridge regression for predicting used car prices, we recommend that used car dealers leverage the power of ridge regression models to better inform their inventory decisions. This will ensure that they are consistently making the most profitable choices.