Cars4U

Exploratory Data Analysis & Price Modeling

PG-DSBA Project 3
Eric Green
Feb 2020

Background

In 2018-19, while new car sales were recorded at 3.6 million units, around 4 million second-hand cars were bought and sold. There is a slowdown in new car sales and that could mean that the demand is shifting towards the pre-owned market. In fact, some car sellers replace their old cars with pre-owned cars instead of buying new ones.

Unlike new cars, where price and supply are fairly deterministic and managed by OEMs (Original Equipment Manufacturer / except for dealership level discounts which come into play only in the last stage of the customer journey), used cars are very different beasts with huge uncertainty in both pricing and supply.

Keeping this in mind, the pricing scheme of these used cars becomes important in order to grow in the market.

Objectives

Come up with a pricing model that can effectively predict the price of used cars and can help the business in devising profitable strategies using differential pricing. For example, if the business knows the market price, it will never sell anything below it.

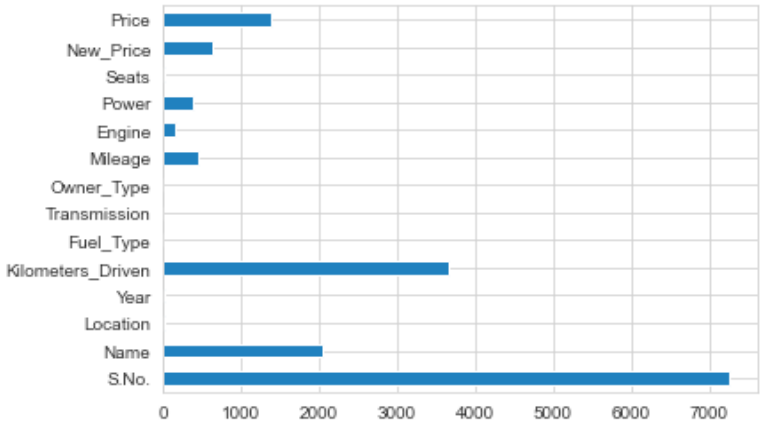
- Explore and visualize the dataset
- Build a linear regression model to predict the prices of used cars
- Generate a set of insights and recommendations that will help the business

Data Summary

Data Dictionary

S.No.: Serial Number
Name: Name of the car which includes Brand name and Model name
Location: The location in which the car is being sold or is available for purchase Cities
Year: Manufacturing year of the car
Kilometers_Driven: The total kilometers driven in the car by the previous owner(s) in KM.
Fuel_Type: The type of fuel used by the car. (Petrol, Diesel, Electric, CNG, LPG)
Transmission: The type of transmission used by the car. (Automatic / Manual)
Owner_Type: Type of ownership
Mileage: The standard mileage offered by the car company in kmpl or km/kg
Engine: The displacement volume of the engine in CC.
Power: The maximum power of the engine in bhp.
Seats: The number of seats in the car.
New_Price: The price of a new car of the same model in INR Lakhs.(1 Lakh = 100, 000)
Price: The price of the used çin INR Lakhs (1 Lakh = 100, 000)

Unique Values



Missing Values

	null values
S.No.	0
Name	0
Location	0
Year	0
Kilometers_Driven	0
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	2
Engine	46
Power	46
Seats	53
New_Price	6247
Price	1234

Observations

The raw data set had several issues needing to be addressed prior to visualization and modeling e.g., missing values, 'null' strings, irrelevant columns and extreme outliers.

1. Columns that do not to need value cleansing:
   - Owner_Type
   - Transmission (convert to categorical, encode)
   - Fuel_Type (convert to categorical, encode)
   - Kilometers_Driven
   - Year
   - Location (convert to categorical, encode)
2. Columns needing cleansing/treatment:
   - Mileage (2 NaN, "kmpl" and "km/kg" string processing, convert to num)
   - Engine (46 NaN, 'CC' string processing, convert to num)
   - Power (46 NaN, "bhp" string processing, convert to num)
   - Seats (53 NaN, convert to num)
   - New_Price (6247 NaN, "lakh" string processing, convert to num)
   - Power (1234 NaN)

Raw Data
Rows: 7253
Columns: 14

Data Description

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
S.No.	7253	NaN	NaN	NaN	3626	2093.91	0	1813	3626	5439	7252
Name	7253	2041	Mahindra XUV500 W8 2WD	55	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Location	7253	11	Mumbai	949	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Year	7253	NaN	NaN	NaN	2013.37	3.25442	1996	2011	2014	2016	2019
Kilometers_Driven	7253	NaN	NaN	NaN	58699.1	84427.7	171	34000	53416	73000	6.5e+06
Fuel_Type	7253	5	Diesel	3852	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Transmission	7253	2	Manual	5204	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Owner_Type	7253	4	First	5952	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Mileage	7251	450	17.0 kmpl	207	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Engine	7207	150	1197 CC	732	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Power	7207	386	74 bhp	280	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Seats	7200	NaN	NaN	NaN	5.27972	0.81166	0	5	5	5	10
New_Price	1006	625	4.78 Lakh	6	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Price	6019	NaN	NaN	NaN	9.47947	11.1879	0.44	3.5	5.64	9.95	160

Data Preprocessing – Stage 1 (column vectors)

Initial Preprocessed Data

INITIAL Preprocessed data

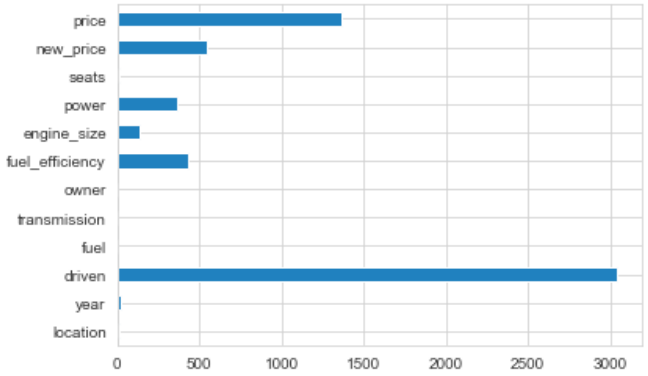
	location	year	driven	fuel	transmission	owner	fuel_efficiency	engine_size	power	seats	new_price	price
0	Mumbai	2010	72000	CNG	man	1	26.60	998.0	58.16	5.0	0.00	1.75
1	Pune	2015	41000	Diesel	man	1	19.67	1582.0	126.20	5.0	0.00	12.50
2	Chennai	2011	46000	Petrol	man	1	18.20	1199.0	88.70	5.0	8.61	4.50
3	Chennai	2012	87000	Diesel	man	1	20.77	1248.0	88.76	7.0	0.00	6.00
4	Coimbatore	2013	40670	Diesel	auto	2	15.20	1968.0	140.80	5.0	0.00	17.74
5	Hyderabad	2012	75000	LPG	man	1	21.10	814.0	55.20	5.0	0.00	2.35
6	Jaipur	2013	86999	Diesel	man	1	23.08	1461.0	63.10	5.0	0.00	3.50
7	Mumbai	2016	36000	Diesel	auto	1	11.36	2755.0	171.50	8.0	21.00	17.50
8	Pune	2013	64430	Diesel	man	1	20.54	1598.0	103.60	5.0	0.00	5.20
9	Chennai	2012	65932	Diesel	man	2	22.30	1248.0	74.00	5.0	0.00	1.95
10	Kochi	2018	25692	Petrol	man	1	21.56	1462.0	103.25	5.0	10.65	9.95
11	Kolkata	2012	60000	Petrol	auto	1	16.80	1497.0	116.30	5.0	0.00	4.49
12	Jaipur	2015	64424	Diesel	man	1	25.20	1248.0	74.00	5.0	0.00	5.60
13	Delhi	2014	72000	Diesel	auto	1	12.70	2179.0	187.70	5.0	0.00	27.00
14	Pune	2012	85000	Diesel	auto	2	0.00	2179.0	115.00	5.0	0.00	17.50
15	Delhi	2014	110000	Diesel	man	1	13.50	2477.0	175.56	7.0	32.01	15.00
16	Kochi	2016	58950	Diesel	man	1	25.80	1498.0	98.60	5.0	0.00	5.40
17	Jaipur	2017	25000	Diesel	man	1	28.40	1248.0	74.00	5.0	0.00	5.99
18	Kochi	2014	77469	Diesel	man	1	20.45	1461.0	83.80	5.0	0.00	6.34
19	Bangalore	2014	78500	Diesel	auto	1	14.84	2143.0	167.62	5.0	0.00	28.00

Info

```
<Cleaned info>
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5872 entries, 0 to 6018
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   location         5872 non-null   category
 1   year             5872 non-null   int64
 2   driven           5872 non-null   int64
 3   fuel             5872 non-null   category
 4   transmission     5872 non-null   category
 5   owner            5872 non-null   int64
 6   fuel_efficiency  5872 non-null   float64
 7   engine_size      5872 non-null   float64
 8   power            5872 non-null   float64
 9   seats            5872 non-null   float64
 10  new_price        5872 non-null   float64
 11  price            5872 non-null   float64
dtypes: category(3), float64(6), int64(3)
memory usage: 476.6 KB
```

Preprocessed
Rows: 5872
Columns: 12

Unique Values



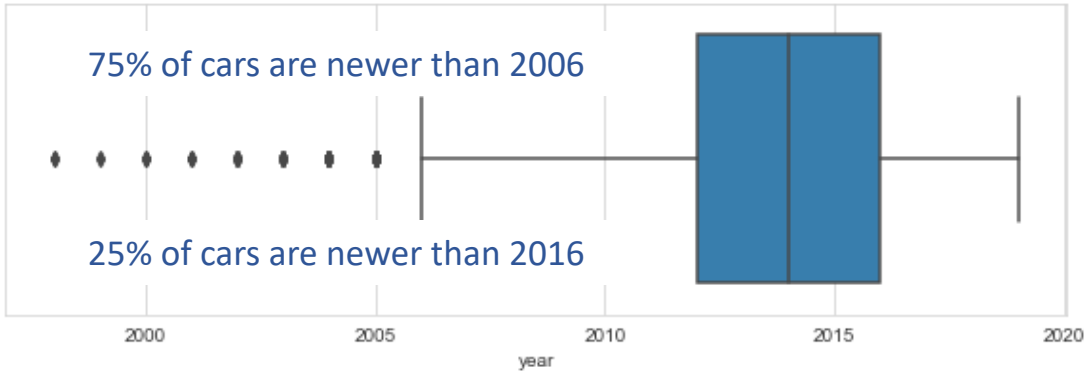

```
<Cleaned uniques>
location            11
year                22
driven            3038
fuel                 4
transmission         2
owner                4
fuel_efficiency    429
engine_size        139
power              368
seats                8
new_price          541
price             1364
dtype: int64
```

Observations

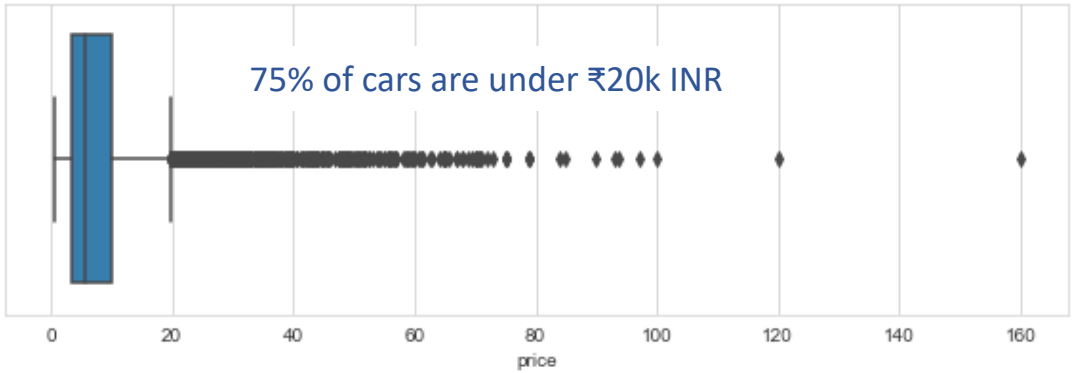
1. Several cleaning methods were applied in initial preprocessing steps
2. Nulls, NAs, null were stripped from the data
3. Number strings were trimmed (units) to produce pure number strings for conversion
4. Columns were renamed and converted to appropriate category or numeric types
5. "price" was initially imputed with median - this did not benefit the R-Squared value
    - Tradeoff decision was made to drop na rows instead of impute dependent variable values - this allowed modeling work to proceed
6. "new price" was imputed with 0, though an explicit use for the 0 was never determined. This is a potential future way to increase model performance
7. This data preprocessing sequence removed 19.04% of the original rows - an intentional tradeoff decision for model accuracy
8. There is still outlier treatments and one hot coding to be done

Univariate EDA

Year



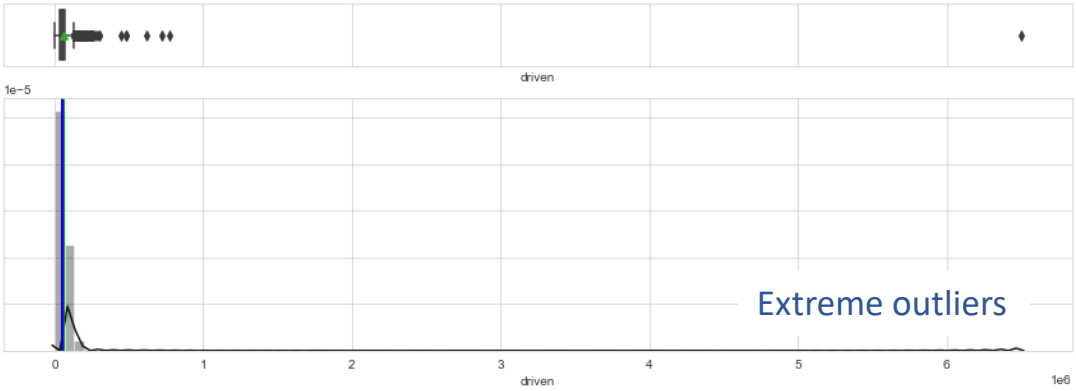
Price (Dependent Variable)



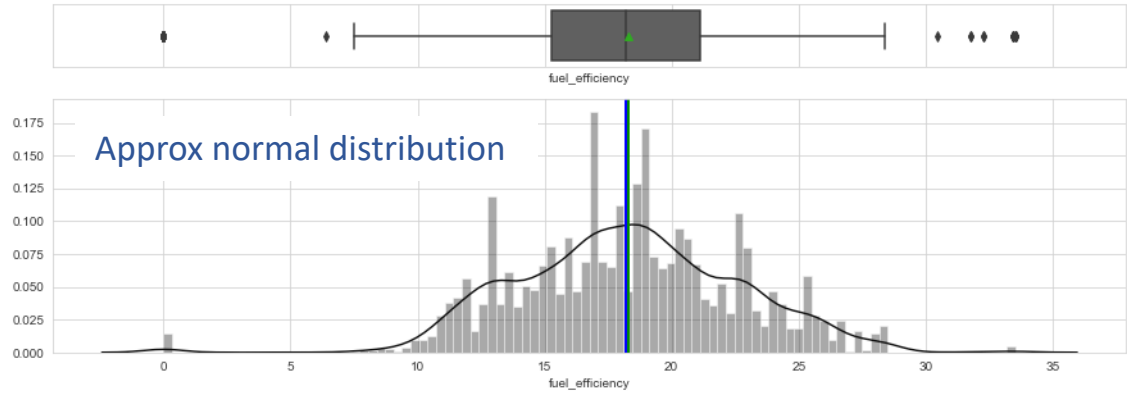
Observations

1. Year distribution is a smooth curve with majority of cars 2010 or newer
2. Driven distribution is extremely right skewed / long tailed. The outliers are dealt with through iterative treatment and testing
3. Fuel efficiency has an approximate normal shape with numbers close to zero potentially problemmatic
4. Engine size is right skewed with outliers
5. Power is right skewed with outliers
6. Seats mode is 5, which is the most common type of car in the sample set
7. Price is right skewed with outliers. These outliers may need to be clipped or otherwise treated to improve model performance

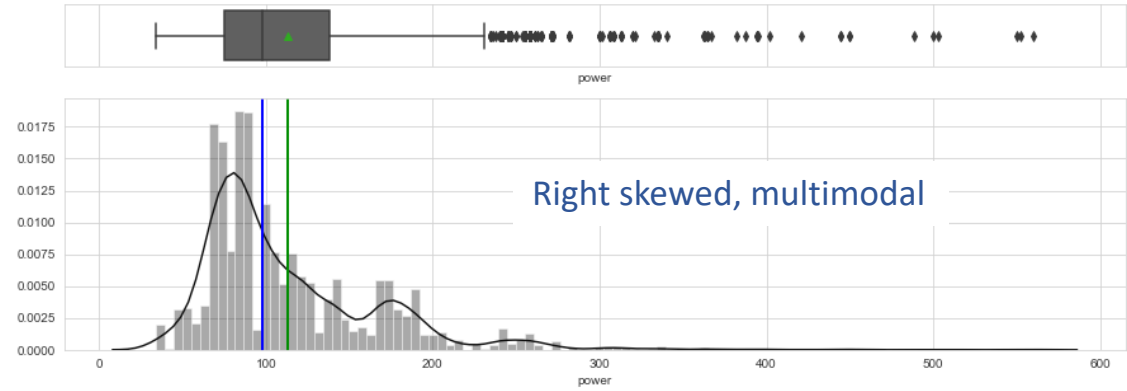
Km Driven



Fuel Efficiency



Power

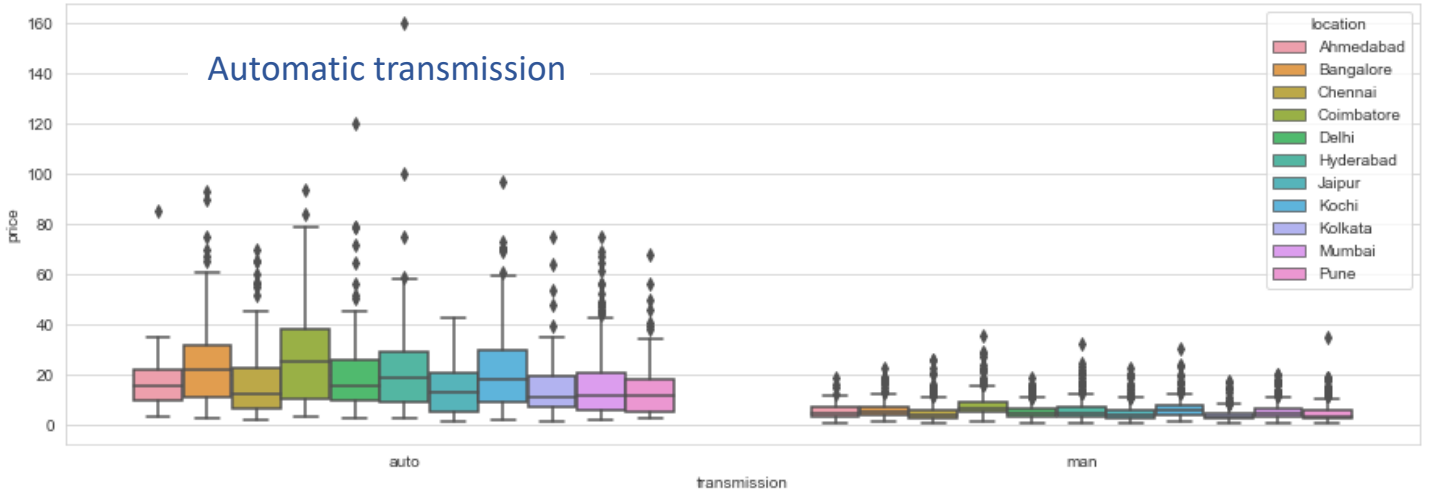


Multivariate EDA

Location & Year vs. Price

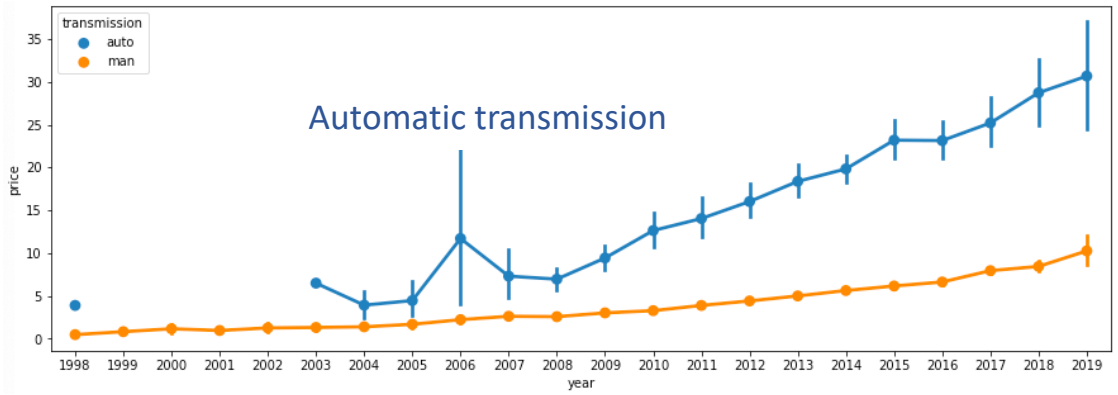
Observation

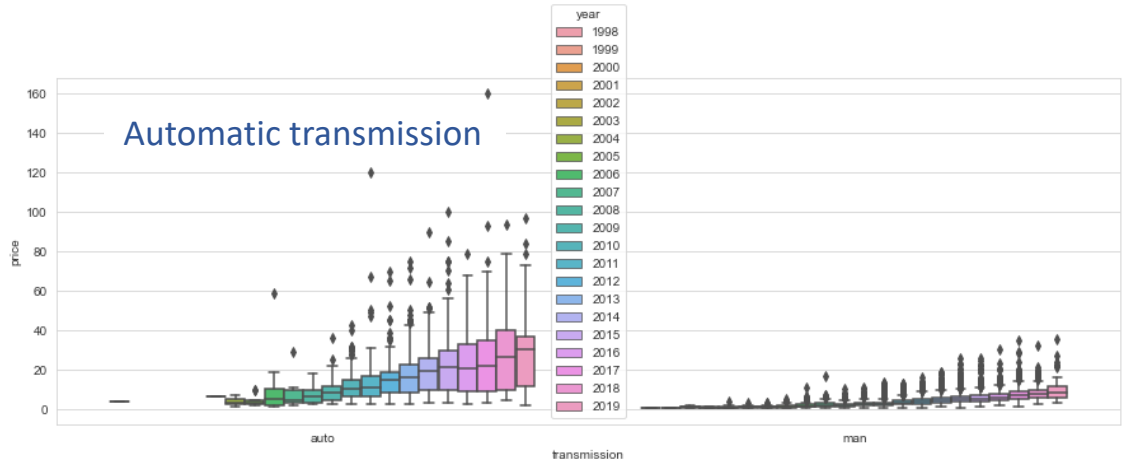
Regardless of location, automatic transmission cars sell for higher prices than manual transmission cars



Observation

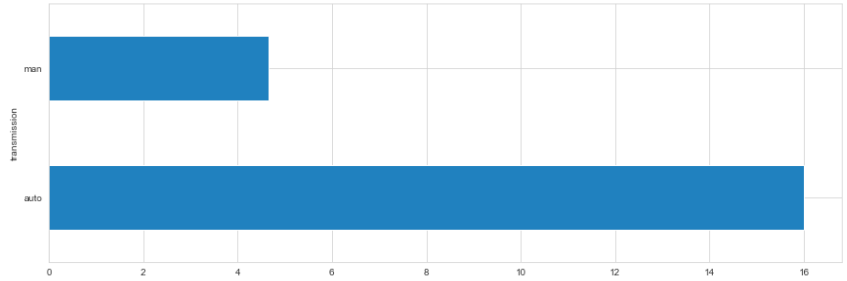
Regardless of year/age, automatic transmission cars sell for higher prices than manual transmission cars

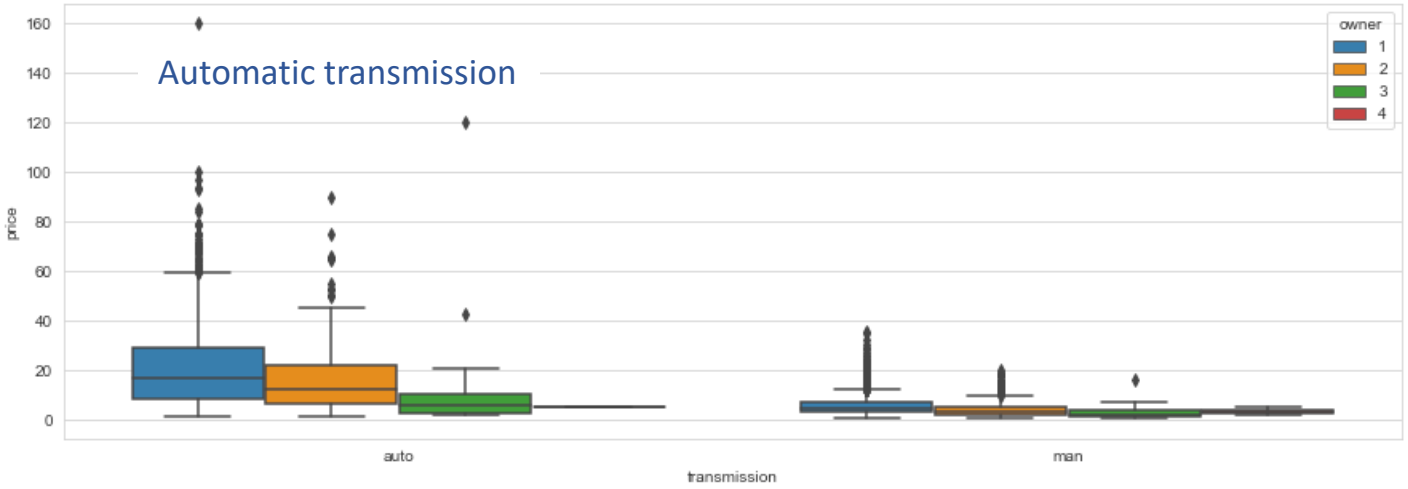




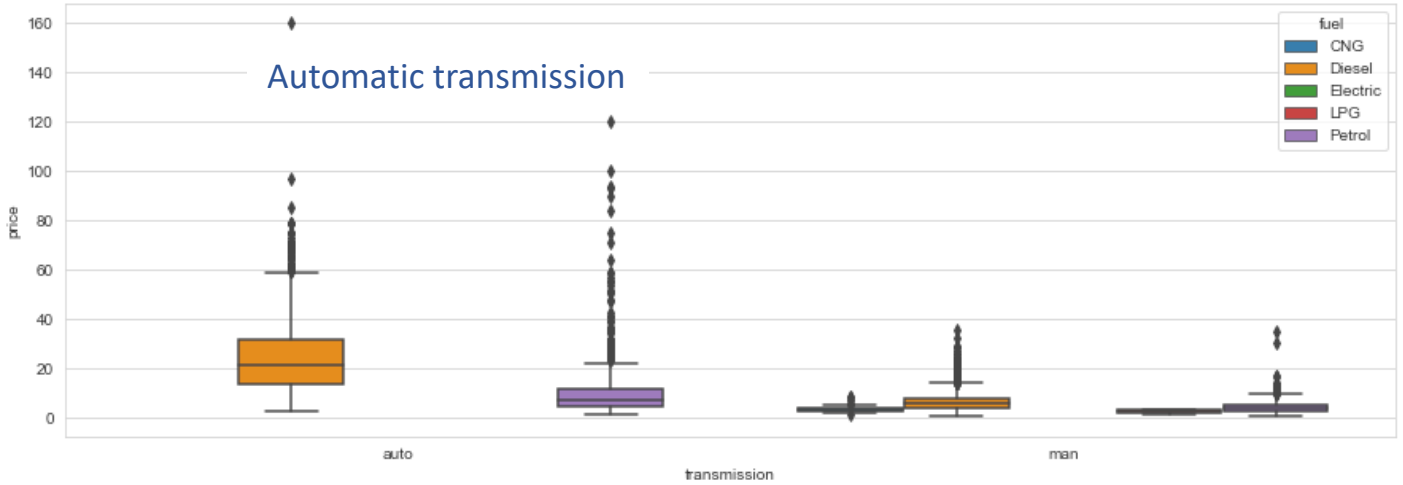
Multivariate EDA

Observation
Median price for automatic transmission is > 3x higher than manual transmission





Observation
Regardless of number of owners, automatic transmission cars sell for higher prices than manual transmission cars

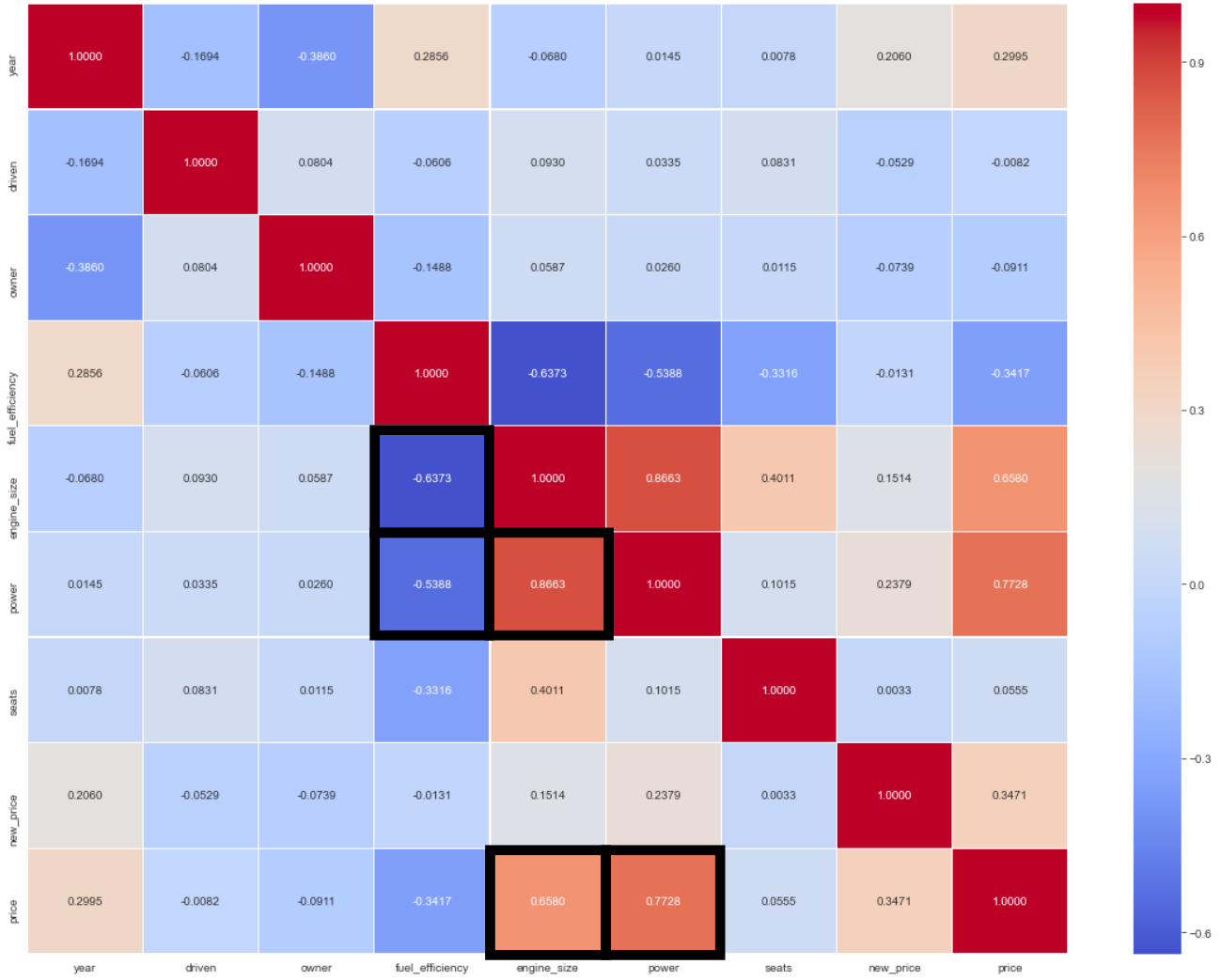


Observation
Regardless of fuel, automatic transmission cars sell for higher prices than manual transmission cars

*It is therefore reasonable to suspect that the transmission variable is important for building a reliable model to predict price

Variable Associations (Initial)

Correlation heatmap allows us to see initial associations between variables. Here we get an initial sense of which variables have collinearity and which variables may be important in modeling and predicting price.

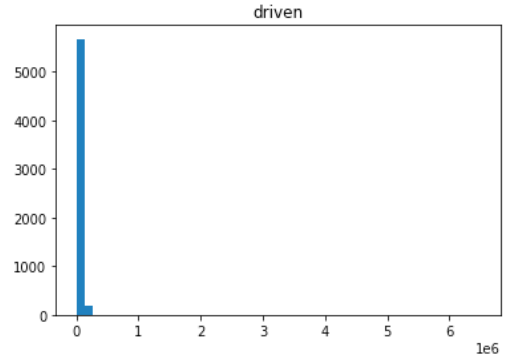


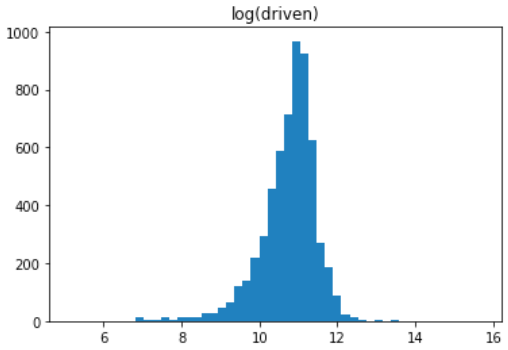
Observations

1. There are associations between
   - (+) power & engine size
   - (+) power & price
   - (+) engine size & price
   - (-) engine eize & fuel efficiency
   - (-) power & fuel efficiency
2. Testing assumptions - all variables but the dependent variable should be independent
   - This is not the case - we should consider removing the colinearity before modeling
   - We can try removing "engine_size" from the data set, since power has a stronger association with price (.704 vs. .601)

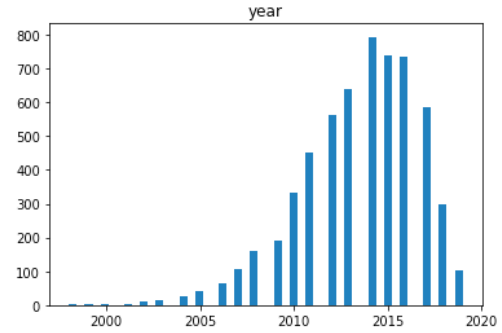
Data Preprocessing – Final (Scaling)

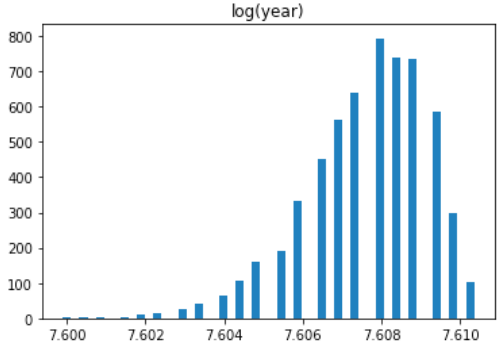
driven (km)



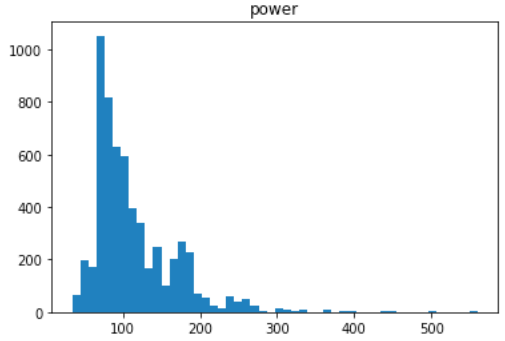


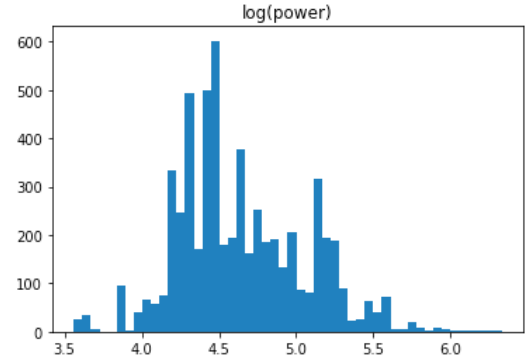
year



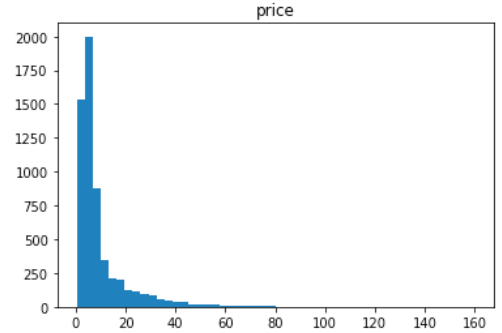


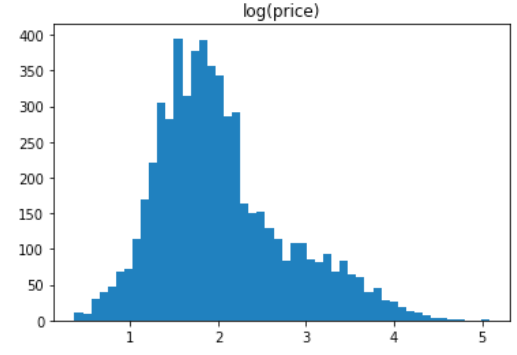
power





price





Observations

The above 4 variables were scaled with log to bring their x-axes to the same order of magnitude and to reduce skewness. Prior to doing this log scaling, the R-Squared value on test data would not go above .7. This proved to be a critical step in achieving an R-Squared value of .88.

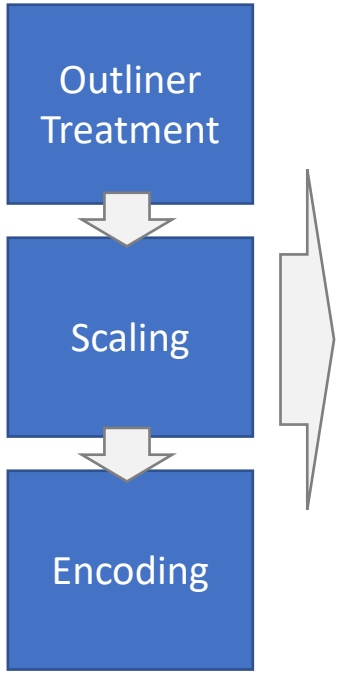
Data Preprocessing – Final

Further Data Preprocessing

Driven and fuel_efficiency outliers were pushed back to upper and lower 1.5 IQR whiskers

Year, driven, power and price were scaling using log to bring their x-axes to be the same order of magnitude

Location, fuel and transmission variables were one hot encoded to enable the model to comprehend them as independent variables




```
Shape (5872, 31)

<class 'pandas.core.frame.DataFrame'>
Int64Index: 5872 entries, 0 to 6018
Data columns (total 31 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   year                 5872 non-null   int64
 1   driven               5872 non-null   float64
 2   owner                5872 non-null   int64
 3   fuel_efficiency      5872 non-null   float64
 4   engine_size          5872 non-null   float64
 5   power                5872 non-null   float64
 6   seats                5872 non-null   float64
 7   new_price            5872 non-null   float64
 8   price                5872 non-null   float64
 9   year_log             5872 non-null   float64
 10  driven_log           5872 non-null   float64
 11  power_log            5872 non-null   float64
 12  price_log            5872 non-null   float64
 13  location_Ahmedabad   5872 non-null   uint8
 14  location_Bangalore   5872 non-null   uint8
 15  location_Chennai     5872 non-null   uint8
 16  location_Coimbatore  5872 non-null   uint8
 17  location_Delhi       5872 non-null   uint8
 18  location_Hyderabad   5872 non-null   uint8
 19  location_Jaipur      5872 non-null   uint8
 20  location_Kochi       5872 non-null   uint8
 21  location_Kolkata     5872 non-null   uint8
 22  location_Mumbai      5872 non-null   uint8
 23  location_Pune        5872 non-null   uint8
 24  fuel_CNG             5872 non-null   uint8
 25  fuel_Diesel          5872 non-null   uint8
 26  fuel_Electric        5872 non-null   uint8
 27  fuel_LPG             5872 non-null   uint8
 28  fuel_Petrol          5872 non-null   uint8
 29  transmission_auto    5872 non-null   uint8
 30  transmission_man     5872 non-null   uint8
dtypes: float64(11), int64(2), uint8(18)
memory usage: 905.5 KB
```

Preprocessed data (final) - ready for modeling

year	driven	owner	fuel_efficiency	engine_size	power	seats	new_price	price	year_log	driven_log	power_log	price_log	location_Ahmedabad	location_Bangalore
2010	72000.0	1	26.60	998.0	58.16	5.0	0.00	1.75	7.605890	11.184421	4.063198	0.559616	0	0
2015	41000.0	1	19.67	1582.0	126.20	5.0	0.00	12.50	7.608374	10.621327	4.837868	2.525729	0	0
2011	46000.0	1	18.20	1199.0	88.70	5.0	8.61	4.50	7.606387	10.736397	4.485260	1.504077	0	0
2012	87000.0	1	20.77	1248.0	88.76	7.0	0.00	6.00	7.606885	11.373663	4.485936	1.791759	0	0
2013	40670.0	2	15.20	1968.0	140.80	5.0	0.00	17.74	7.607381	10.613246	4.947340	2.875822	0	0
2012	75000.0	1	21.10	814.0	55.20	5.0	0.00	2.35	7.606885	11.225243	4.010963	0.854415	0	0
2013	86999.0	1	23.08	1461.0	63.10	5.0	0.00	3.50	7.607381	11.373652	4.144721	1.252763	0	0
2016	36000.0	1	11.36	2755.0	171.50	8.0	21.00	17.50	7.608871	10.491274	5.144583	2.862201	0	0
2013	64430.0	1	20.54	1598.0	103.60	5.0	0.00	5.20	7.607381	11.073335	4.640537	1.648659	0	0
2012	65932.0	2	22.30	1248.0	74.00	5.0	0.00	1.95	7.606885	11.096379	4.304065	0.667829	0	0
2018	25692.0	1	21.56	1462.0	103.25	5.0	10.65	9.95	7.609862	10.153935	4.637153	2.297573	0	0
2012	60000.0	1	16.80	1497.0	116.30	5.0	0.00	4.49	7.606885	11.002100	4.756173	1.501853	0	0
2015	64424.0	1	25.20	1248.0	74.00	5.0	0.00	5.60	7.608374	11.073242	4.304065	1.722767	0	0
2014	72000.0	1	12.70	2179.0	187.70	5.0	0.00	27.00	7.607878	11.184421	5.234845	3.295837	0	0
2012	85000.0	2	6.50	2179.0	115.00	5.0	0.00	17.50	7.606885	11.350407	4.744932	2.862201	0	0
2014	110000.0	1	13.50	2477.0	175.56	7.0	32.01	15.00	7.607878	11.608236	5.167981	2.708050	0	0
2016	58950.0	1	25.80	1498.0	98.60	5.0	0.00	5.40	7.608871	10.984445	4.591071	1.686399	0	0
2017	25000.0	1	28.40	1248.0	74.00	5.0	0.00	5.99	7.609367	10.126631	4.304065	1.790091	0	0
2014	77469.0	1	20.45	1461.0	83.80	5.0	0.00	6.34	7.607878	11.257633	4.428433	1.846879	0	0
2014	78500.0	1	14.84	2143.0	167.62	5.0	0.00	28.00	7.607878	11.270854	5.121700	3.332205	0	1

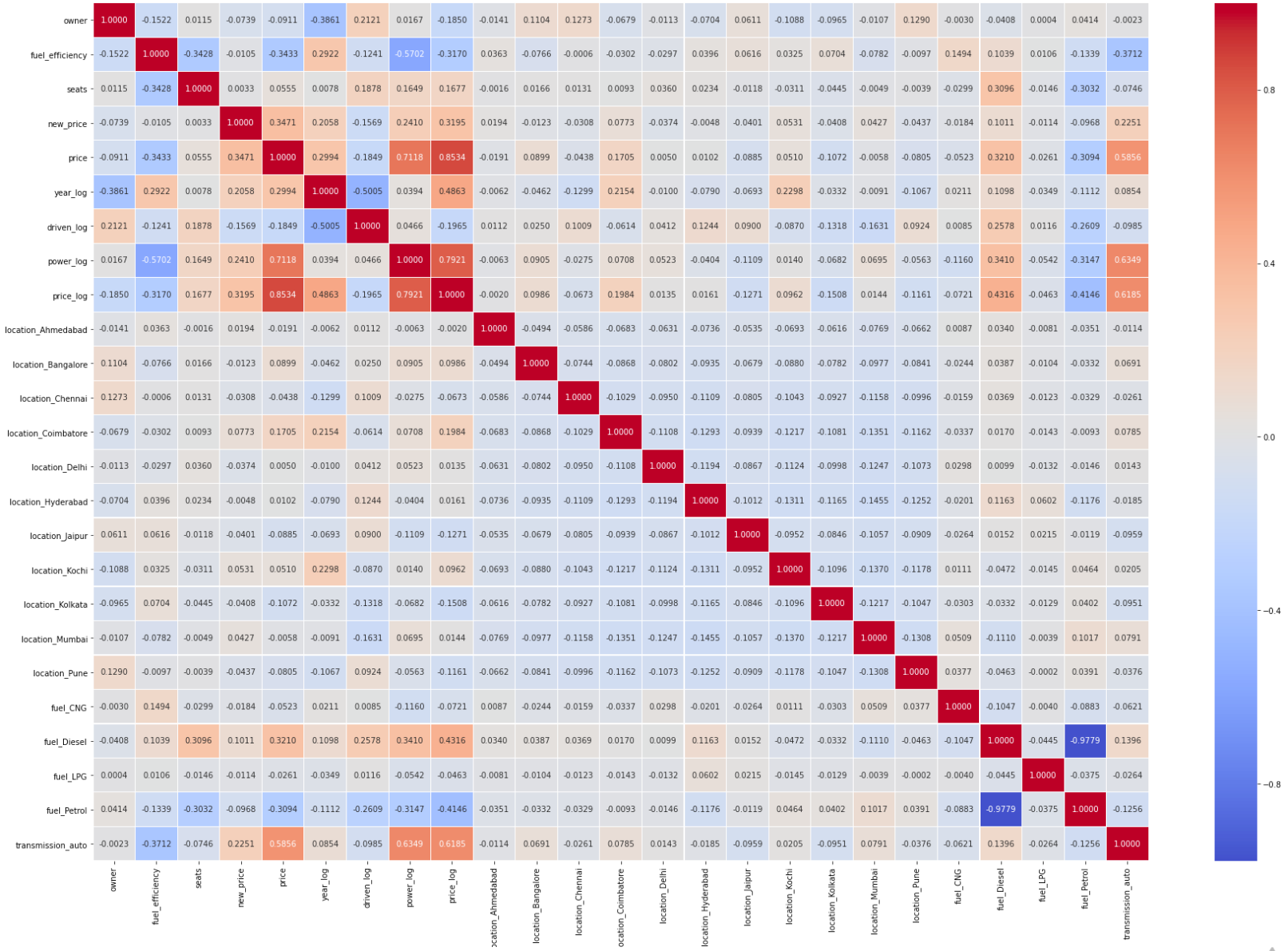
Observations

- Based on exploratory data analysis, the final stage of data preprocessing was executed to realize a data structure suitable for building and test a linear model.
- The final data shape is 5872 rows by 31 columns, including dummy variables

Feature Engineering

Notes on final correlation matrix

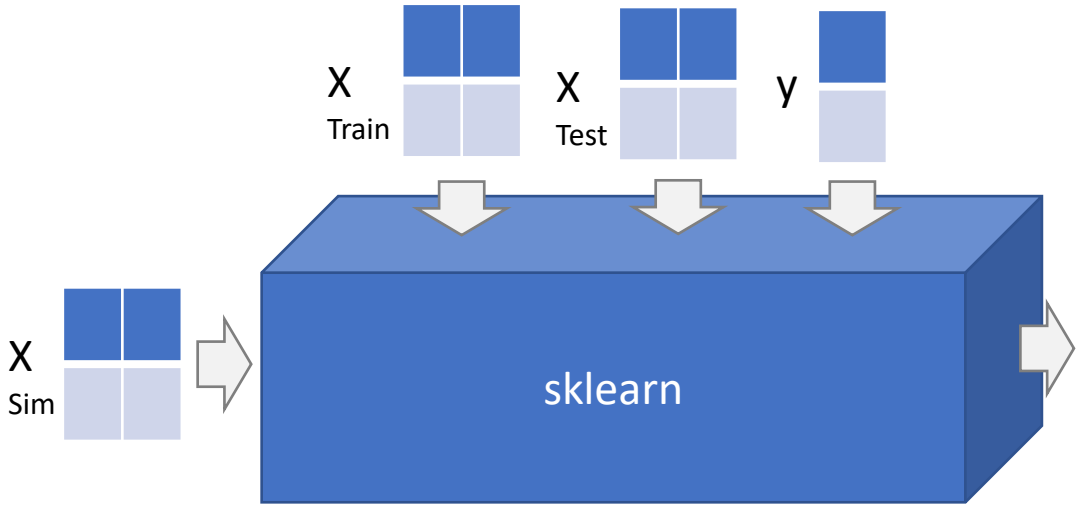
- 4 new log scaled variables are present (driven, year, power, price)
- One hot encoded features were added corresponding to the all location, fuel and transmission categorical values
- Manuel transmission category was removed as it is not needed due to being mutually exclusive with automatic transmission
- engine_size feature was removed due to collinearity with power (power was retained because it has a stronger correlation with price at .7)
- There remain some areas of collinearity between fuel_Diesel and fuel_Petrol, power_log and transmission_auto which should be addressed in future model improvements



sklearn Linear Regression (Price)

Simulated user car

```
# config
year = 2017
owner = 2
seats = 5
fuel_efficiency = 25.5
km_driven = 25
power = 140

used_car = {
'owner': [owner],
'fuel_efficiency': [fuel_efficiency],
'seats': [seats],
'year_log': [np.log(year)],
'driven_log': [np.log(km_driven)],
'power_log': [np.log(power)],
'location_Ahmedabad': [0],
'location_Bangalore': [0],
'location_Chennai': [1],
'location_Coimbatore': [0],
'location_Delhi': [0],
'location_Hyderabad': [0],
'location_Jaipur': [0],
'location_Kochi': [0],
'location_Kolkata': [0],
'location_Mumbai': [0],
'location_Pune': [0],
'fuel_CNG': [0],
'fuel_Diesel': [1],
'fuel_LPG': [0],
'fuel_Petrol': [0],
'transmission_auto': [1]}
```



******** Prediction ********

Simulated base Price before markup:

₹382,8431 INR or $53,598 USD

Model Coefficients

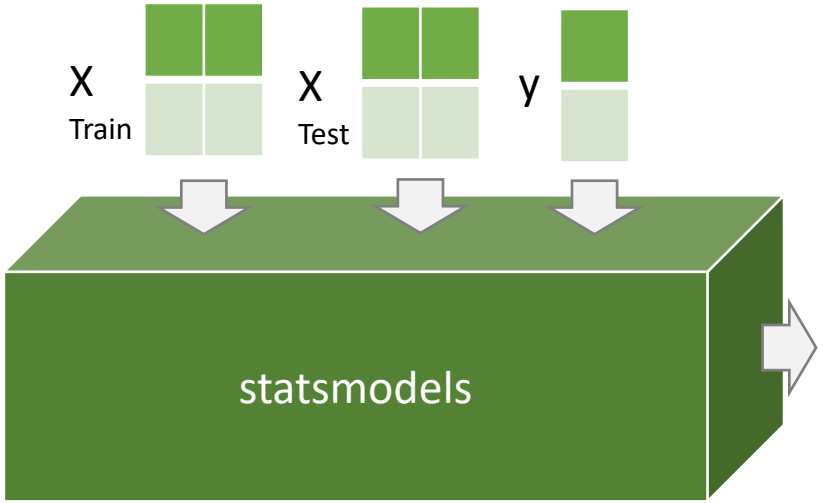
```
Coefficient of owner is: -0.064
Coefficient of fuel_efficiency is: -0.018
Coefficient of seats is: -0.003
Coefficient of year_log is: 213.394
Coefficient of driven_log is: -0.115
Coefficient of power_log is: 1.2
Coefficient of location_Ahmedabad is: 0.006
Coefficient of location_Bangalore is: 0.147
Coefficient of location_Chennai is: 0.033
Coefficient of location_Coimbatore is: 0.117
Coefficient of location_Delhi is: -0.046
Coefficient of location_Hyderabad is: 0.142
Coefficient of location_Jaipur is: -0.029
Coefficient of location_Kochi is: -0.009
Coefficient of location_Kolkata is: -0.235
Coefficient of location_Mumbai is: -0.081
Coefficient of location_Pune is: -0.045
Coefficient of fuel_CNG is: 0.082
Coefficient of fuel_Diesel is: 0.119
Coefficient of fuel_LPG is: 0.014
Coefficient of fuel_Petrol is: -0.215
Coefficient of transmission_auto is: 0.274

Model intercept is: -1625.5223184827069

R-Squared value (train): 0.894
```

Observations

1. Using 70/30 train/test split with sklearn linear regression
2. After trying several conbinations of features and scaling treatments
   - The highest R-Squared value realized on test data was: .88 (initial test R-Square value was .53)
   - Outliers were clipped at 1.5 IQR
   - Feature were treated with log scaling to unify feature x-axes order of magnitude
   - One hot encoding was performed to create numeric features for model comprehension
3. A simulated car object was created and fed to the model for prediction of a price
   - Manipulating the feature inputs moved the price up and down as would be expected
   - More testing and tuning is required here to realize production use robustness
4. The logic could be built into a pricing engine that would calculate base prices and differential factors based on a subset of features
   - Recommend use to determine what to pay for a car (inventory) as well as what to sell it for
   - Recommend use to update car prices in inventory over time
   - As the regression model sees more and more uses cars, the pricing should be come more accurate (representative of all use cars) over time
5. Overall, the model seems to appropriately comprehend linear relationship between the independent variables and the dependent variable, price

R-Squared value (test) .88

statsmodels Linear Regression (Price)



Observations

1. statsmodels uses OLS (ordinary least squares) to fit the linear model
2. Using 70/30 train/test split with statsmodels linear regression
3. R-Squared value achieved .894 by trying different variations of features and scaling, started at .52
4. coefs look to all be similar order of magnitude
5. p-values which are relatively large indicating features that maybe not be significant (>.05) are:
   - location_Ahmedabad
   - location_Chennai
   - location_Delhi
   - location_Jaipur
   - location_Kochi
   - location_Mumbai

   *This may be because the locations are sparsely represented in the test data (needs further investigation)
6. Removing one of the encoded location dummies does not increase the R-Squared value
7. Given the small delta between the R-Squared values of Train vs. Test data, there does not appear to be an over or underfitting problem

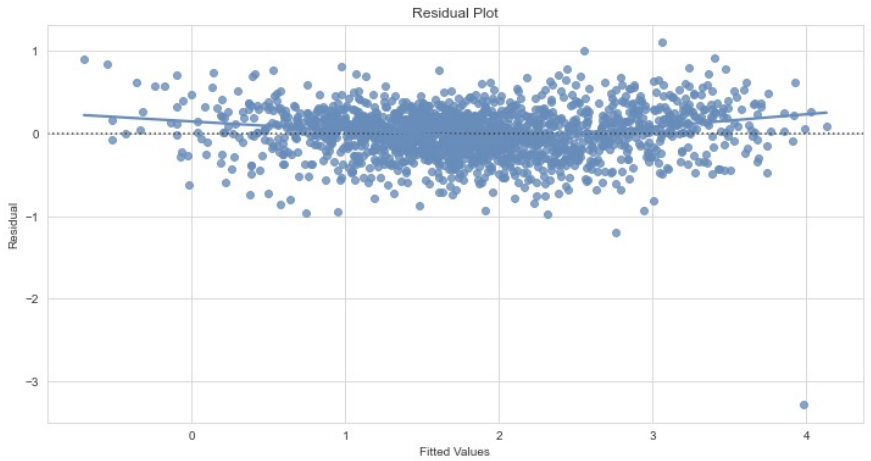
```
*** Model Testing ***

                            OLS Regression Results
==============================================================================
Dep. Variable:              price_log   R-squared:                       0.883
Model:                            OLS   Adj. R-squared:                  0.882
Method:                 Least Squares   F-statistic:                     657.1
Date:                Fri, 26 Feb 2021   Prob (F-statistic):               0.00
Time:                        11:00:42   Log-Likelihood:                -340.22
No. Observations:                1762   AIC:                             722.4
Df Residuals:                    1741   BIC:                             837.4
Df Model:                          20
Covariance Type:            nonrobust
==============================================================================================
                                 coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                      -1348.8122     38.503    -35.032      0.000   -1424.329   -1273.296
owner                         -0.0774      0.018     -4.268      0.000      -0.113      -0.042
fuel_efficiency               -0.0157      0.003     -5.756      0.000      -0.021      -0.010
seats                          0.0276      0.011      2.553      0.011       0.006       0.049
year_log                     221.3085      6.324     34.996      0.000     208.906     233.712
driven_log                    -0.0789      0.012     -6.363      0.000      -0.103      -0.055
power_log                      1.1162      0.029     38.248      0.000       1.059       1.173
location_Ahmedabad             0.0689      0.043      1.606      0.109      -0.015       0.153
location_Bangalore             0.2581      0.038      6.857      0.000       0.184       0.332
location_Chennai               0.0604      0.034      1.786      0.074      -0.006       0.127
location_Coimbatore            0.1817      0.032      5.594      0.000       0.118       0.245
location_Delhi                 0.0039      0.032      0.123      0.902      -0.059       0.067
location_Hyderabad             0.1889      0.031      6.181      0.000       0.129       0.249
location_Jaipur                0.0017      0.037      0.046      0.964      -0.071       0.074
location_Kochi                 0.0276      0.031      0.889      0.374      -0.033       0.089
location_Kolkata              -0.1942      0.033     -5.868      0.000      -0.259      -0.129
location_Mumbai               -0.0212      0.030     -0.704      0.482      -0.080       0.038
fuel_CNG                    -337.1255      9.628    -35.016      0.000    -356.008    -318.243
fuel_Diesel                 -337.0689      9.627    -35.012      0.000    -355.951    -318.187
fuel_LPG                    -337.2328      9.623    -35.044      0.000    -356.107    -318.359
fuel_Petrol                 -337.3850      9.626    -35.048      0.000    -356.265    -318.505
transmission_auto              0.3129      0.021     14.618      0.000       0.271       0.355
==============================================================================
Omnibus:                      481.093   Durbin-Watson:                   1.956
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6201.959
Skew:                          -0.906   Prob(JB):                         0.00
Kurtosis:                      12.011   Cond. No.                     2.47e+16
```

R-Squared value (test) .883

Model Performance

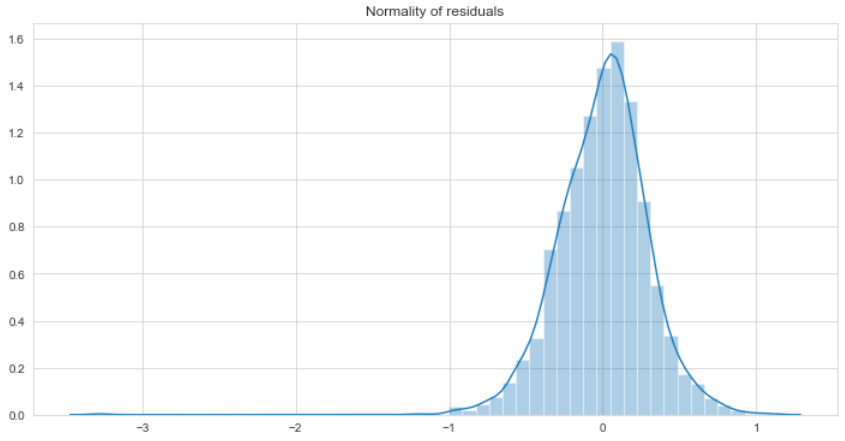
Observation
Residuals show no patterns and appear as white noise

Test for linearity



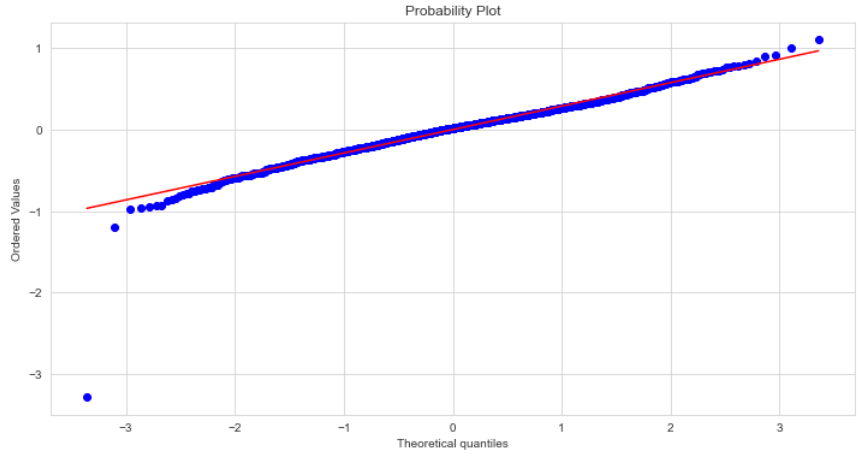
Observation
Distribution of residuals is normally distributed around zero

Test for normality



Observation
The plot line of residuals fit the value prediction line well with slight tapering on the ends

Test for normality



Conclusions

1. Raw data on use cars had significant hygiene issues that had to be addressed to be able to do the visual analysis or regression modeling
2. NA, NaN, 'null' values are removed from the data across various columns
3. Numberic strings had ascii characters stripped out and were converted to number vectors
4. After an unfruitful attempt to impute median for Price nan values, NaN rows were dropped for dependent variable Price
    - This reduced the data size by 19.04% (intentional tradeoff made)
5. EDA revealed year and transmission type to be important in predicting price
6. Initial correlation matrix (heatmap) showed collinearity, subsequently removed by dropping engine_size and keeping power variable due to a stronger association with price
7. Final preprocessing clipped outliers at 1.5 IRQ for driven and fuel_efficiency
8. year, driven, power and price were log scaled to bring them all on the same scale of observation (this had the single biggest impact on model performance)
9. Encoded dummies were generated for categorical variables location, fuel and transmission
10. sklearn linear regression model was fit and tested against training and testing data
    - Through iterative manipulation of features, R-Square value was increased from .53 to .88 (we are happy with this result)
    - A prediction simulator was implemented which predicts car prices given an input used car
    - This pricing simulator can to determine base car prices to sell at, how to update prices on inventory and what prices to buy more inventory at (margin % below base predicted price). Differential pricing factors can be implemented on the base price given the business strategy.
11. statsmodels OLS linear regression model was fit and tested against training and testing data
    - R-Squared value tracks closely with sklearn, final value .883 (adjusted R-Squared .882)
    - No problems with over or underfitting given small delta b/t RS and Adj-RS
12. There remains lingering collinearity in the dummy variables that can be addressed in future to explore better model performance
13. Visual analysis and testing of residuals indicates an effective regression model
    - Residual scatter plot shows no patterns and appears as white noise
    - Distribution of residuals in normally distributed around 0
    - Residuals line fits the predicted values line well (with exception of tappering on the ends)

Recommendations to Business

Recommendations

1. Develop SWOT analysis of used car market by location in India
2. Based on SWOT analysis, define/update company strategy and risks related to price and related factors
    - Define strategy to target differential pricing (by location, transmission, power, km)
3. Prepare data for used car inventories by location
4. Use sklearn regression model code to determine base prices for all inventory
    - Create new price guidance and distribute to all locations
    - Update inventory pricing at all locations
    - Document price ranges to buy new inventory at based on target margin
5. Develop differential pricing model that complements the base price prediction model and implement this at all locations
6. Systemmatically incorporate new used car data into the model to improve pricing and margin targeting longer-term
7. Quarterly, re-evaluate pricing model and update pricing guidance to locations

Further Analysis

1. Lingering collinearity in dummy variables can be reduced to potentially further optimize the pricing model
2. Enhance the pricing model to natively incorporate differential pricing by factor
3. Explore expanding the data parameters collected on use cars. By adding new data paramaters for the model to comprehend, this is a potentially area of innovation and competative advantage (enhanced design, cleanliness and implementation of the raw data itself)