

# Cars4U Pricing Guide

Price Models for Used Vehicles

May 2021



# 01

## **Introduction**

Introduces the scope of the study & major assumptions

# 02

## **Key Findings**

Outlines the major findings of the study and highlights key recommendations

# 03

## **Industry Data**

Describes the key variables examined from a sample of recently sold used cars

# 04

## **Methodology**

Outlines how the linear regression model was constructed and how it can be applied as a business tool

# 05

## **Recommendations**

Demonstrates how to apply the key findings of the study to improve business strategy

# 06

## **Conclusions**

Here you could describe the topic of the section



# 01

## **Introduction**

# The Opportunity



**Used car sales  
outpace new  
cars sales for  
the first time**

A recent slowdown in new car sales has opened the door for the pre-owned car market. As of 2018-19, the pre-owned car market has surpassed the new car market with sales of 4 million to 3.6 million units, respectively. Statistics suggest that some car sellers are replacing their old cars with pre-owned cars instead of buying new ones!

**PROSPECTS**



**Prospective  
shoppers will  
not overpay  
for a used car**

## The Problem



Unlike new cars, pricing pre-owned cars comes with a number of challenges. Original Equipment Manufacturers (OEMs) determine the price and supply of new vehicles, but there is no regulation of either price or supply in the used car market. This creates great uncertainty surrounding supply and demand and confusion regarding setting prices.



# The Solution



Using historical data from recent used car sales, we can construct a pricing model that can effectively predict the price of a future sales of used cars. This can help Cars4U to device profitable strategies using differential pricing.



**If Cars4U knows  
the market value,  
it will never sell a  
car for anything  
less!**

**COMPETITIVE EDGE**

# Assumption 1

For the purposes of this study, the two unique units of Mileage found in the dataset (kmpl and km/kg) are assumed to be equivalent. The units may be used interchangeably and, when applying our model to find the price of a used car, either of the two units may be used in the calculation process.

# Assumption 2

The dataset listed currency in both Lakh and Crore. Lakh is defined as 100,000 Indian Rupees and Crore is defined as 100 Lakh. To standardize currency, Price and New Price variables were converted to USD at the exchange rate of 1 INR = 0.013 USD, which was current as of April 28, 2021.



# Limitation #1

One limitation of the equation is that all 6 of the variables of the vehicle need to be known or else it will not adequately reflect the price of the car. For example, if the Power of the car is not known and a sales rep enters 0, the car will be underpriced by almost \$185 per unit of power. The mean of Power is 112 bhp, so leaving power blank on a car which has close to a mean value for its power, would artificially drop the price by almost \$21,000. In a lot of cases, the model would give a price that is negative!

# Limitation #2

The equation will not work well on vehicles that have variables do not match the any of the categorical options in the model. For example, trying to price a Subaru with this model would require guessing a similar brand name and would not be accurate. Similarly, pricing a car outside of the 10 city options would be problematic.



# Limitation #3

The equation is only as good as its underlying dataset. While the data has been thoroughly cleaned, small human error input mistakes would not be flagged during the cleaning process and could cause increased variation to the Price variable, lowering the reliability of the model. For example, accidentally inputting 31,000,000 for a car worth 13,000 would be caught, but accidentally inputting 31,000 for the same car would probably go unnoticed.





# 02

## **Key Findings**



# 76.2%

**The computer trained model explains 76.2% of  
the variations of the dependent variable, Y.**



## The equation that achieves the best fit is:

$$\begin{aligned} \text{Price} = & -2,062,000 + 1029.59[\text{Years}] - 0.0127[\text{KM Driven}] - 83.52[\text{Mileage}] + 118.74[\text{Power}] - \\ & 13,560.00[\text{Hyundai}] - 13,640.00[\text{Isuzu}] + 8,037.28[\text{Jaguar}] - 9,864.59[\text{Jeep}] + \\ & 82,440.00[\text{Lamborghini}] + 18,360.00[\text{Land Rover}] - 13,710.00[\text{Mahindra}] - 12,060.00[\text{Maruti}] + \\ & 1,771.70[\text{Mercedes-Benz}] + 5,532.68[\text{Mini}] - 11,010.00[\text{Mitsubishi}] - 13,170.00[\text{Nissan}] + \\ & 13,880.00[\text{Porsche}] - 13,670.00[\text{Renault}] - 13,960.00[\text{Skoda}] - 14,180.00[\text{Tata}] - 9,154.71[\text{Toyota}] - \\ & 13,670.00[\text{Volkswagen}] - 7,272.09[\text{Volvo}] + 1,952.92[\text{Bangalore}] + 1,158.98[\text{Chennai}] + \\ & 1,793.26[\text{Coimbatore}] - 1,056.68[\text{Delhi}] + 1,997.84[\text{Hyderabad}] + 1,168.54[\text{Jaipur}] - 574.31[\text{Kochi}] - \\ & 1,286.16[\text{Kolkata}] - 1,227.45[\text{Mumbai}] + 414.56[\text{Pune}] \end{aligned}$$

## PRICE FORMULA

The above equation looks more complicated than it is. There are essentially only 6 variables, because when choosing to fit this price model to a particular car, only 1 name and 1 location can be selected. For example, if you look up a Hyundai in Bangalore, then all other names and locations are filled with a 0 value and it wipes out those coefficients from the equation, since any value multiplied by 0 becomes 0 itself. See next slide for clarification...

## Randomly Selecting an individual car:

**2016 Toyota, 24,000k, 100bpm, 12.8kmpl in Chennai**

$$\text{PRICE} = -2,062,000 + 1029.59[2016] - 0.0127[24,000] - 83.52[12.8] + 118.74[100] - 9,154.71[1] + 1,158.98[1]$$

**PRICE FORMULA**

Let's apply the formula on the previous slide to the following selection: A 2016 Toyota with 24,000km, 100bpm of power, 12.8kmpl for mileage in the city of Chennai. The equation on the previous slide is now reduced to the simplified equation above. Since Toyota is selected, all other Names receive 0s and cancel out their corresponding coefficients. The same occurs when Chennai was selected for the location variables. The equation is now much simpler to understand and apply to specific automobiles.

1

Power has the highest individual correlation with Price. When run individual, Power explains more of the variation of the dependent Y variable (Price) than any other independent X variable.



2

After categorizing the Names into Brand Names, this categorical data is the second-best indicator at explaining variations in the Price data.



3

The year of the car is correlated with price. Newer models command higher resale prices. The model's year is the third best indicator at explaining Price variations.

4

Location also seems to explain the Price variations, since the same type of car can command higher prices in some cities compared to others, all else being equal.



5

Kilometers Driven also helps explain the Price variation. Its moderate, negative correlation with Price shows that as the car accumulates more kilometers, its value drops.

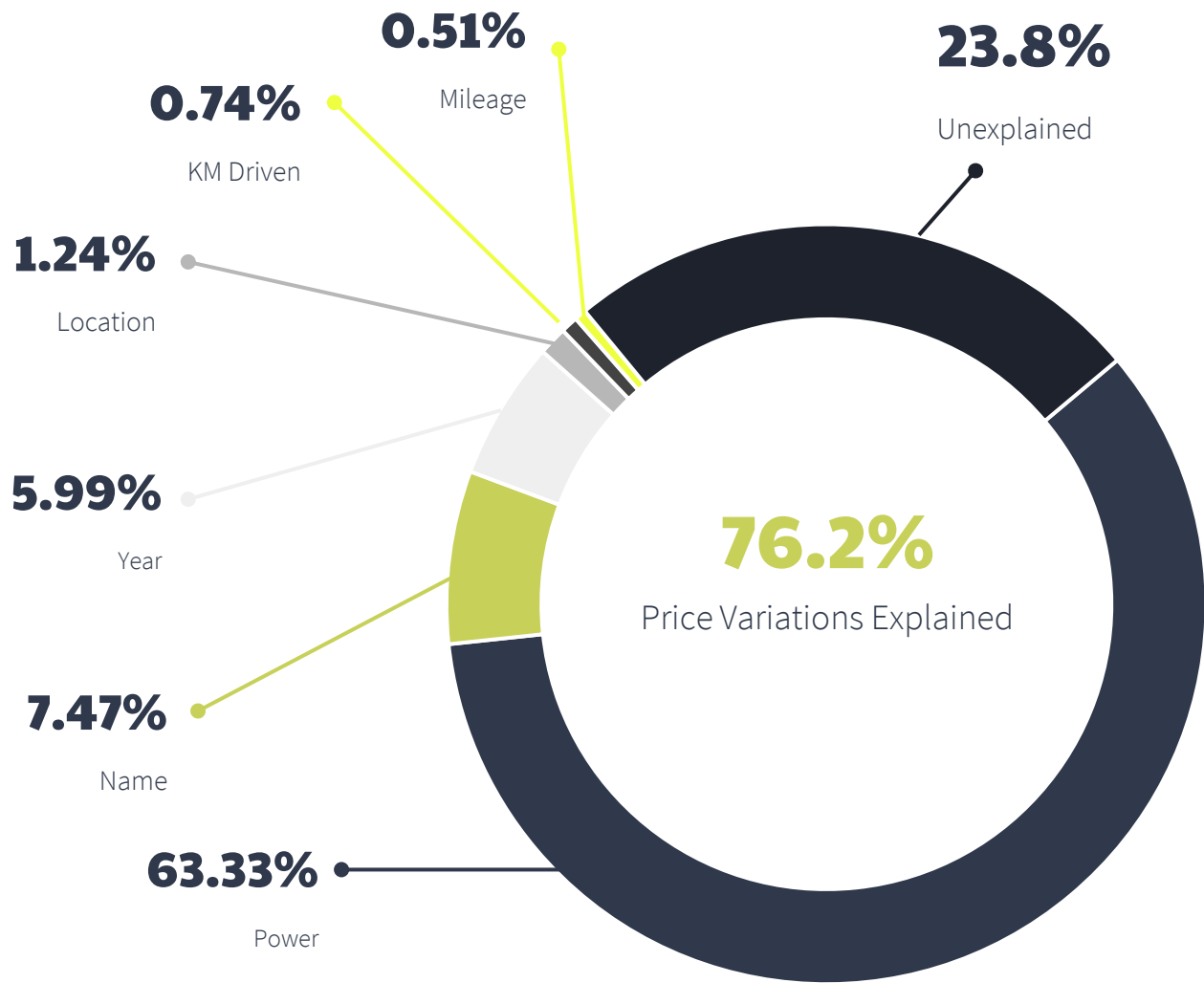


6

Mileage is the weakest indicator of the key variables, but it does still add value. Mileage is negatively correlated to Price: as fuel efficiency drops, the price increases.



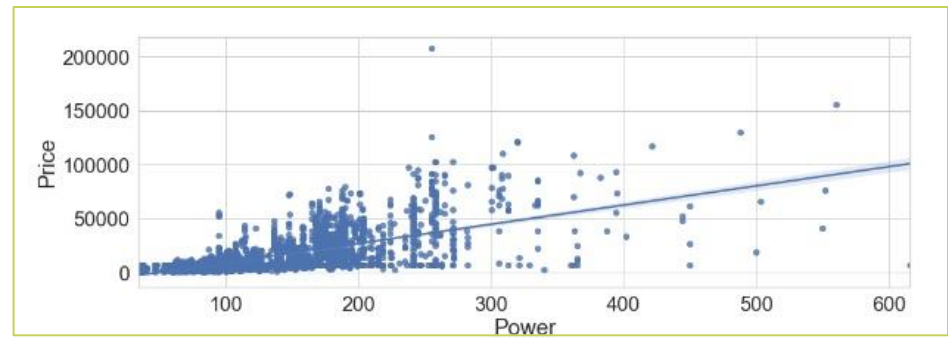
## AFFECTS OF THE VARIABLES



Starting with the Power variable, which had the highest correlation with Price, we added one variable at a time to view the increase in the R-squared value, which is a measure of the variation of price. Each of these variables increased the R-squared value by the respective percentages shown in the graph. The cumulative effect of these 6 variables explains 76.2% of all Price variations. The addition of other variables failed to improve the R-squared value any further and were excluded from the final model.

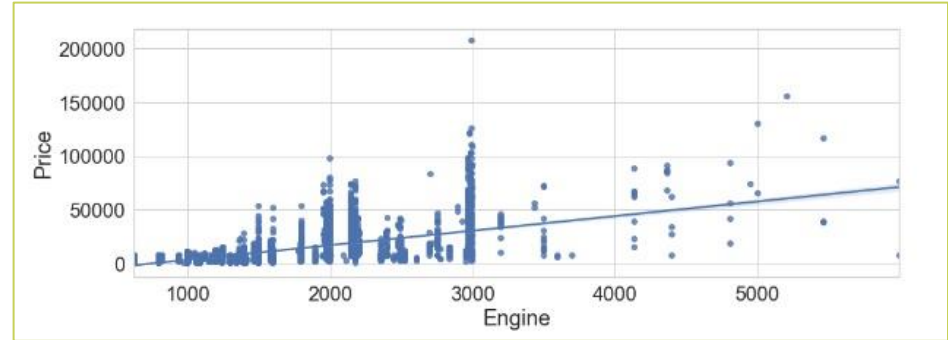
Price is most closely correlated with Power (0.70). The positive correlation means that as the power of the vehicle increases, customers are willing to pay more for the car.

1



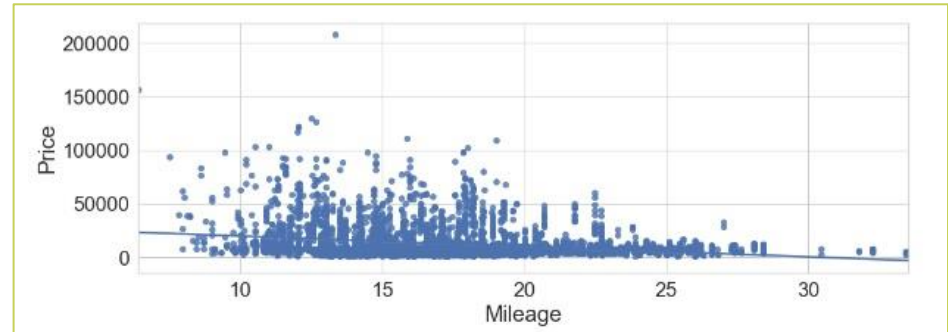
Price is also strongly correlated with Engine (0.60). This positive correlation means that as the engine size increases, so does the price of the car, all other things being equal. Engine showed high multicollinearity with Power so was ultimately removed from the final model.

2



There is a moderately negative correlation between Price and Mileage. As mileage improves, the price of the car decreases.

3



TOP CORRELATIONS WITH PRICE

# NO CORRELATION WITH PRICE

## SERIAL NUMBER

The unique Serial Number was not a determining factor when developing the price model

-0.02

## SEATS

The number of seats was studied but did not affect the price model with any significance

-0.05

## OWNER TYPE

The number of previous owners was examined and found not to influence the price model significantly

-0.10

**NAME**

The inclusion of Name helped increase the amount of Price variation by almost 7.5%

**7.47%****LOCATION**

Location increased the percentage of Price variation explained by the model by 1.24%

**1.24%****FUEL TYPE**

Electric, LPG and CNG fuel types were underrepresented in the sample and Petrol and Diesel caused multicollinearity. The inclusion of this variable increased value added by less than 0.1%. It was therefore excluded to help simplify the model

**<0.1%****TRANSMISSION**

The p-value of Transmission was too big to be considered significant. When included, it dropped the explanation of Price variance by 0.1%, so was excluded.

**-0.1%**



# 03

## **Industry Data**



## BRAND NAME



30 distinct automobile brands representing manufacturers on 4 continents

## LOCATION



10 locations across Indian, representing all four geographical regions (N,S,E,W)

## FUEL TYPE\*



4 different fuel types: Petrol, Diesel, CNG, LPG

## YEAR



Spanning almost 25 years,: 1996 and 1998 to 2019, inclusive

\* Fuel Type initially added value to the regression analysis, but Petrol and Diesel showed high multicollinearity and so the variable was removed from the final model.

## KILOMETERS DRIVEN



Ranging from almost new (<200km) to industry workhorses (>700,000km)

## MILEAGE



From gas-guzzling Lamborghinis (<6km/L) to fuel efficient Marutis (33+km/L)

## ENGINE\*



Spanning the tiny Tata (624 CC) to the monstrous Bentley (5998 CC)

## POWER



Empomassing the gentle Maruti (34.2 bhp) to the powerful Bentley (616 bhp)

\* On its own, Engine initially added value to the regression analysis, but it showed high multicollinearity with Power and so the variable was removed from the final model since Power provided a higher R-squared value.

## SERIAL NUMBER

The serial number was a randomly assigned and unique identification number.



## TRANSMISSION\*

The dataset contained two types of transmission: Manual or Automatic



## OWNER\*

The Owner Type indicated how many previous owners the vehicle had, ranging from 0 to 4+



## SEATS\*

Describes how many seats the vehicle had. Discrete values included 2 and 4-10




\* These variables were included in regression analysis but did not add value to the model, so are deemed non-relevant variables and excluded from the final model.

## DEPENDENT VARIABLE

Price shows strong, positive correlations with the Engine and Power variable. It also shows moderate, negative correlations with Mileage and Year. Customers will pay more for bigger, less fuel-efficient engines which provides more power, usually found in luxury/sports models. As the car ages, customers are not willing to pay as much.

## INDEPENDENT VARIABLES

The independent variables in this study have varying degrees of correlation with one another. -1.0 denotes perfect negative correlation, 1.0 is perfect positive correlation. For this study, we define -0.09 to 0.09 as no correlation, 0.10 to 0.24 as a weak, positive correlation, 0.25 to 0.59 as a moderate, positive correlation, and 0.60 to 0.99 as a strong, positive correlation. Similarly, on the negative side, -0.09 to -0.24, -0.25 to -0.59 and -0.60 to -0.99 represent weak, moderate and strong negative relationships, respectively. See following slides for addition info...

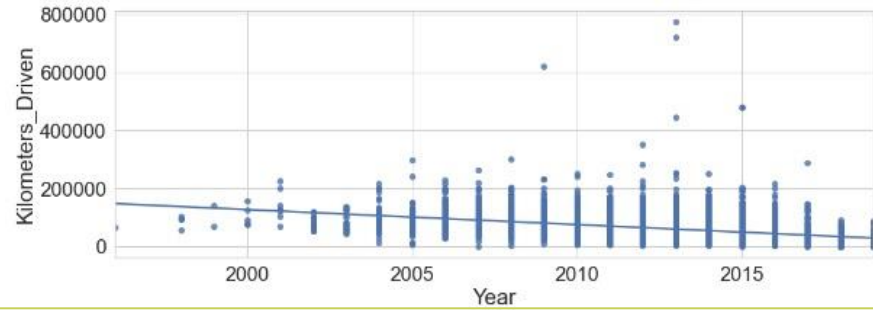


	Price
Year	0.276030
KM/Driven	-0.154741
Mileage	-0.299160
Engine	0.601946
Power	0.702710
Seats	0.048075
Price	1.000000



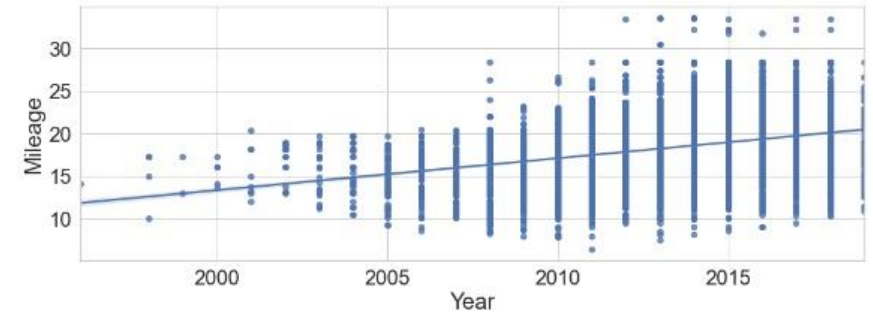
Year and Kilometers Driven are moderately negatively correlated (-0.45). Newer model cars (ie. Higher years are more recent) have not had the time to accumulate as many kilometers as older used cars.

1



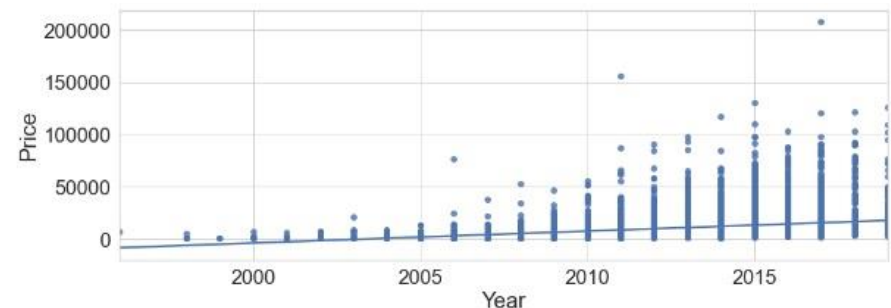
Year and Mileage are moderately positively correlated, (0.30). This is logical because, as technology improves, mileage should become more efficient

2



Year and Price have a close to moderate positive correlation (0.28). People are willing to pay more for newer used cars, presumably, because they are in better condition.

3

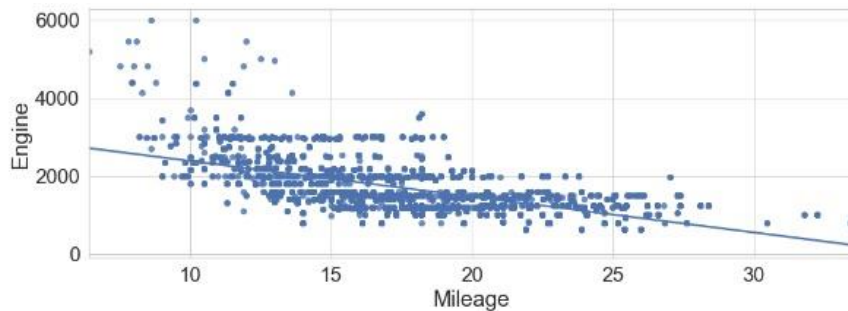


YEAR



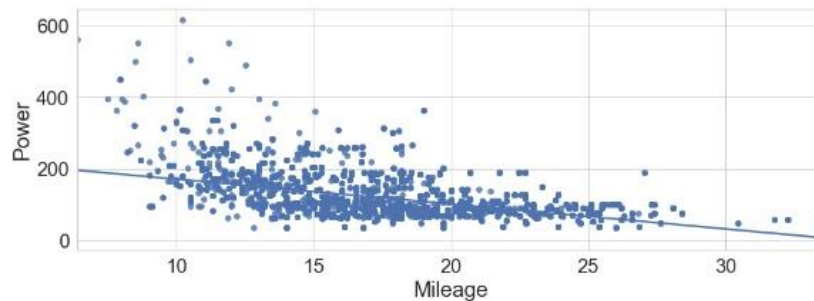
Mileage has a strong, negative correlation with engine size (-0.64). As engines grow bigger, mileage is less efficient and goes down.

1



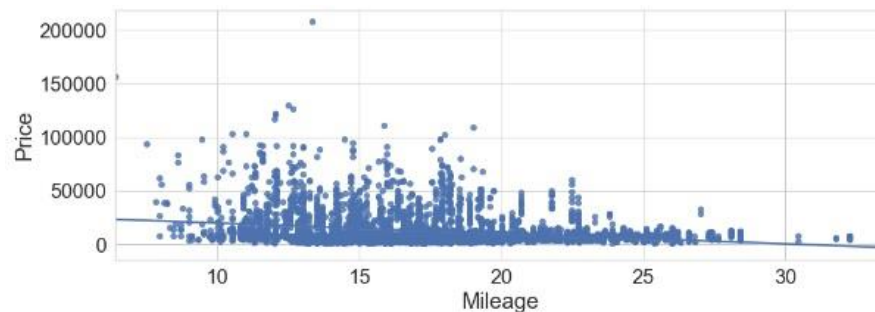
Mileage has a moderately negative correlation with Power (-0.54). As power increases, it requires more fuel and thus mileage goes down.

2



Mileage has a negative correlation with Price, although it is moderate (-0.3). It could be that fuel efficient cars are cheaper, or that luxury brand cars have lower fuel efficiency.

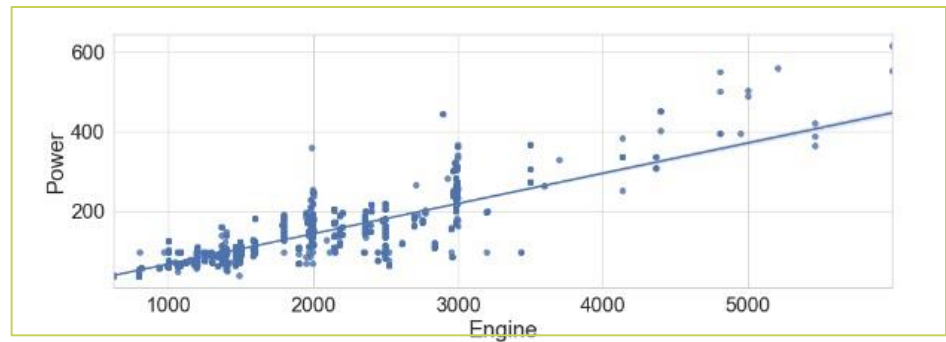
3



**MILEAGE**

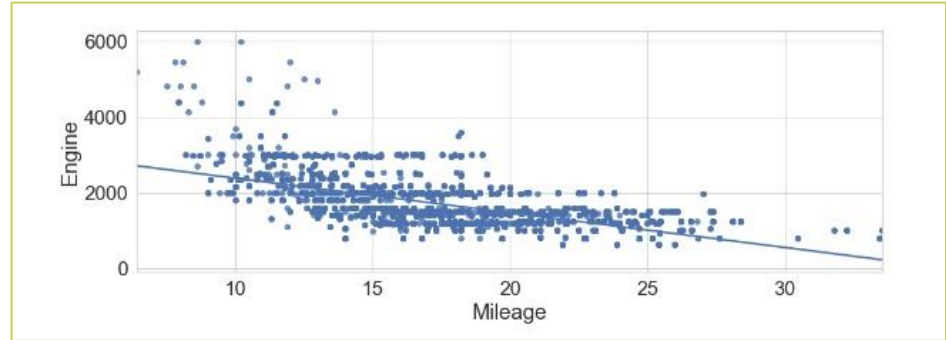
Engine is most strongly correlated with Power (0.85). Bigger engines tend to produce more power and the luxury/sports vehicles score high in both categories.

1



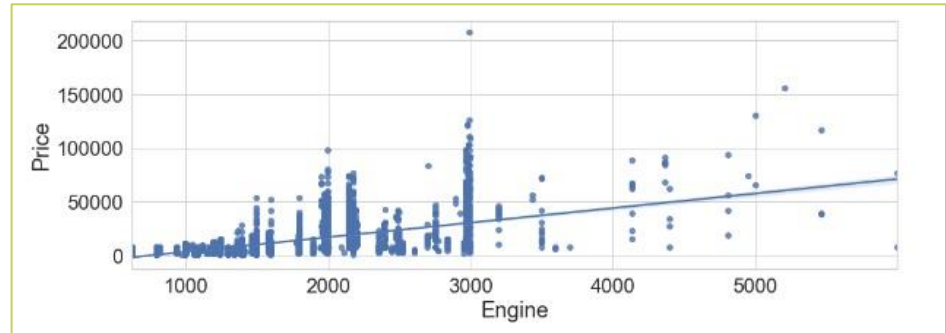
Engine has a strong, negative correlation with Mileage (-0.64). Bigger engines are less fuel efficient, requiring more fuel per kilometer to function.

2



Engine has a strong, positive relation to Price (0.60). People seem to be willing to pay more for a bigger engine. Bigger engines are a feature of luxury/sports models.

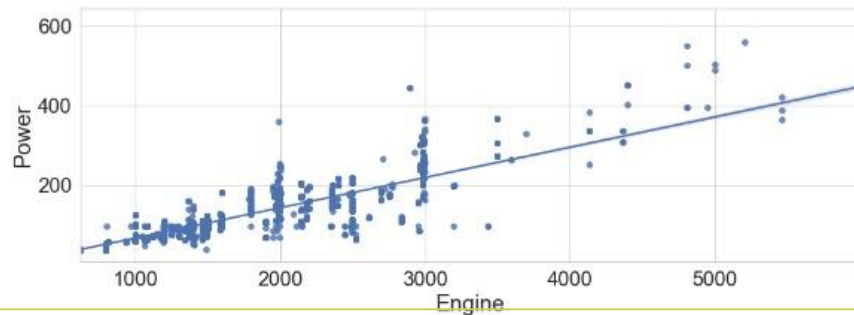
3



ENGINE

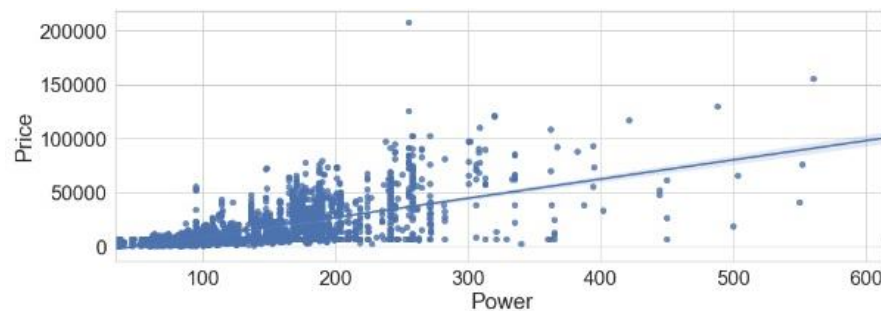
Power is strongly correlated with engine (0.85). This positive correlation makes sense because larger engines produce more power.

1



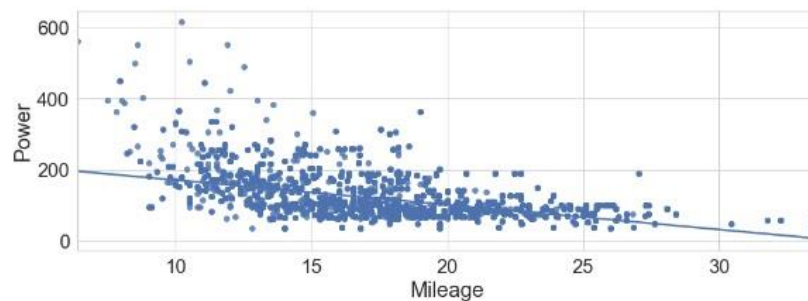
Power is also strongly correlated to Price (0.70). More powerful cars are simply more expensive to purchase.

2



Power is moderately, negatively correlated to Mileage (-0.54). More powerful cars are less fuel efficient than less powerful ones.

3



POWER



# 04

## **Methodology**

# METHODOLOGY

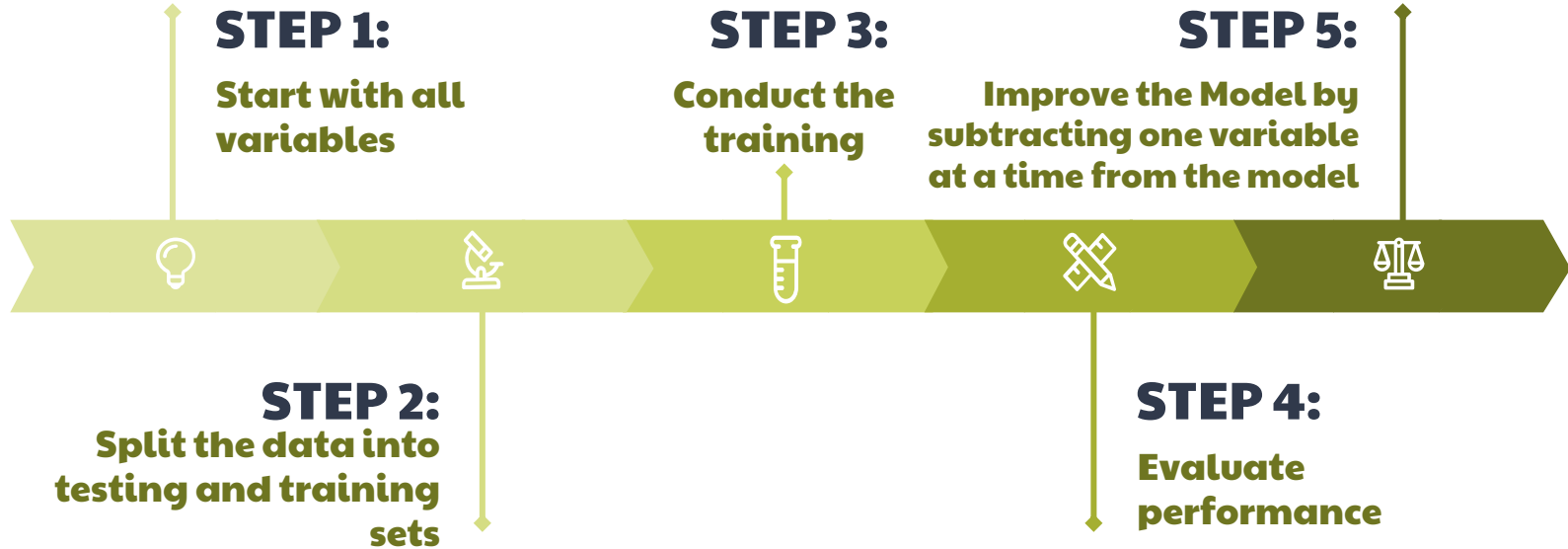
Linear Regression analysis can only be applied to data that is first cleaned and free of missing values. To clean the data, each independent variable was studied to identify its individual characteristics, including whether it is categorical, continuous or discrete. Variables then had to be converted to a standardized unit, for example, New Price was converted from Crore and Lakh to USD. Certain variables, such as Serial Number, were deemed non-essential and thus removed from the model. Missing values were dropped from the data set, replaced with variable means or medians where appropriate. For example, Fuel Type = 'Electric' was dropped from the dataset because there were only two entries, and both had missing values under the Price column. Name was converted to 31 distinct categorical groups by brand name. Price, which had 1234 missing values, was cleaned by replacing null values with grouped medians determined by brand name. For example, a row where Name = 'Toyota' that had a missing value, took the price median of all Toyotas. This method was used to increase accuracy over replacing missing price values with the column median and median was chosen over mean because the variable was skewed.

# METHODOLOGY

## SELECTING THE MODEL

Once the data was cleaned, we chose a model that splits the data into two groups: training and testing data. This method provides the computer with a target to aim for, when constructing the model, and is considered Supervised Learning. One limitation with this method, is it cannot interpret categorical variables. Therefore, each distinct value in each categorical variable had to be split into individual columns using the `pd.get_dummies` function in python. Transmission, for example, was converted to two distinct columns: Manual and Automatic. If the row had previously contained Manual under the Transmission column, it now contained a '1', meaning it was 'turned on'. The new Automatic column would now contain a '0', corresponding to being 'turned off'. This method prevents the program from placing more importance on higher values had we recoded Manual = 1, Automatic = 2. The data was split in a 70:30 ratio between training data and testing data and regression analysis was performed. The R-squared value was then evaluated to understand how much of the variation in the Price data is explained by our model. The model was tested using the statsmodels package to determine whether we could improve the model. Finally, the model was re-run using "Forward Feature Selection" to see if we could improve our equation by adding, instead of subtracting, one variable at a time. The process use for these two methods are outlined in the following slides...

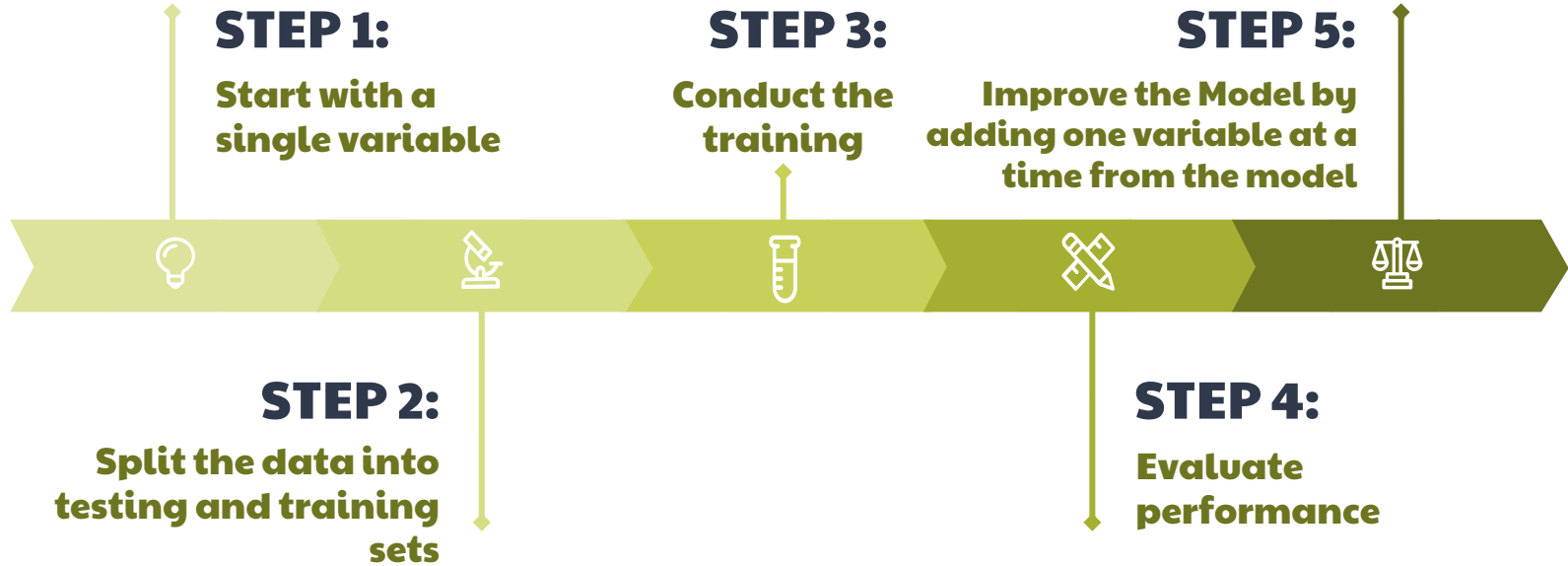
# METHODOLOGY



REMOVING VARIABLES METHOD



# METHODOLOGY



ADDING VARIABLES METHOD

# SUBTRACTING VARIABLES METHOD

## STEP 1: Start with ALL the Variables (Create dummy variables for categorical data)

Regression analysis cannot be performed on categorical values. Instead, categorical columns need to be transformed into dummy variables. Each unique variable is converted to a new column and is 'turned on' in its respective row but 'turned off' in the other columns created from the original categorical data.



```
X = pd.get_dummies(X, columns=['Name', 'Location', 'Fuel_Type', 'Transmission', 'Owner_Type'], drop_first=True)
X.head()
```

	index	Year	Kilometers_Driven	Mileage	Engine	Power	Seats	Name_BMW	Name_Bentley	Name_Chevrolet	...	Location_Kolkata	Location_Mumbai	Locz
0	0	2010	72000	26.60	998.0	58.16	5.0	0	0	0	...	0	1	
1	1	2015	41000	19.67	1582.0	126.20	5.0	0	0	0	...	0	0	
2	2	2011	46000	18.20	1199.0	88.70	5.0	0	0	0	...	0	0	
3	3	2012	87000	20.77	1248.0	88.76	7.0	0	0	0	...	0	0	
4	4	2013	40670	15.20	1968.0	140.80	5.0	0	0	0	...	0	0	

5 rows × 54 columns

## STEP 2: Split into Test (30%) and Training (70%)



```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

# SUBTRACTING VARIABLES METHOD

## STEP 3: Conduct the Training

Python will perform regression analysis on the training model and output coefficients for the variables used in the model. This includes the newly created dummy variables.



```
Coefficients of the equation are: [[-1.33441815e-01  9.98683511e+02 -2.37384860e-02 -1.69353807e+02
 2.00853552e+00  8.35655901e+01  4.19491395e+02 -1.47713287e+03
 9.29776422e+03 -1.60745647e+04 -1.72844240e+04 -1.51347981e+04
-1.69184272e+04 -1.39957288e+04  9.34414857e-09 -1.53564359e+04
-1.44756836e+04 -1.43266559e+04  5.11984147e+03 -1.06197350e+04
 9.02321795e+04  2.07346876e+04 -1.75813068e+04 -1.35014276e+04
 5.25186371e+02  5.22679502e+03 -1.49210647e+04 -1.53224248e+04
 2.88491719e-09  2.01410902e+04 -1.50847069e+04 -1.47157718e+04
-1.06047082e-09 -1.59690767e+04 -1.22815332e+04 -1.53913272e+04
-8.15689136e+03  1.80712413e+03  1.79416335e+03  1.84020639e+03
-8.78385302e+02  1.57103483e+03  1.04829581e+03 -3.41346802e+02
-1.46385254e+03 -1.15681119e+03  4.50429493e+02  9.37820552e+01
 6.23515340e+02 -1.36470120e+03  1.28007850e+02  1.39057524e+03
-4.41505055e+02  5.19537877e+01]]
```

## STEP 4: Evaluate Performance



```
The Mean Absolute Error on the test is 3842.7606312230223
The Root Mean Absolute Error on the test is 6927.747168449878
The R-squared value on the test is 0.7484903835397203
```

# SUBTRACTING VARIABLES METHOD

## STEP 5: Improve the Model

Either manually or using a python package, we can **SUBTRACT** one variable at a time to see the affect it has on the overall model. Statsmodels is a python package that helps show multicollinearity and variables with low p-values so that the variable that is contributing the least value (or negative value) is removed from the model first. The model is re-run and the process is repeated until the adjusted R-squared value begins to lower. At this point, the previous model was the best.



Fuel_Type_Diesel	31.166010
Fuel_Type_LPG	1.705411
Fuel_Type_Petrol	32.169239
Transmission_Manual	2.255530

Step 5a: Partial output of Fuel\_Types (above) show vif scores >10, meaning that Diesel and Petrol have multicollinearity. Since they are categorical variables of the same category (Fuel\_Type) the whole category was removed.

Step 5b: Partial output of OLS regression (right) shows the R-squared values we are analyzing as well as p-values >0.05 for both Mileage and Seats. Since Mileage is higher, we remove this variable first and then repeat.

OLS Regression Results		
=====		
R-squared:		0.765
Adj. R-squared:		0.763
F-statistic:	340.8	
Prob (F-statistic):	0.00	
Log-likelihood:	-52001.	
AIC:	1.041e+05	
BIC:	1.044e+05	
=====		
	coef	P> t
const	-2.027e+06	0.000
Year	1010.7447	0.000
Kilometers_Driven	-0.0172	0.000
Mileage	-27.4288	0.440
Engine	3.3893	0.000
Power	89.4992	0.000
Seats	174.1506	0.347

# ADDING VARIABLES METHOD

## STEP 1: Start with ONE the Variable

Take the independent variable that has the strongest correlation with the dependent variable. In this case, Power had the strongest correlation with Price.



*#Defining X and y variables*

```
X = UCD[['Power']]  
Y = UCD[['Price']]
```

## STEP 2: Split into Test (30%) and Training (70%)



```
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_state=42)
```

# ADDING VARIABLES METHOD

## STEP 3: Conduct the Training

**This method works in reverse of the subtracting method. We start with a very simple model: the dependent variable and one independent variable. The result is that  $\text{Price} = -10,326.96 + (199.22)\text{Power}$ .**



Intercept of the linear equation: `[-10326.95719222]`

Coefficients of the equation are: `[[199.22604974]]`

## STEP 4: Evaluate Performance



The Mean Absolute Error on the test is `5273.091740290603`

The Root Mean Absolute Error on the test is `8699.868228993011`

The R-squared value on the test is `0.6033605699772082`

**The MAE and RMSE are much higher when you start with one variable, but go down as more relevant variables are added. Conversely, the model only predicts 60% of price variations but will increase as more relevant variables are added.**

# SUBTRACTING VARIABLES METHOD

## STEP 5: Improve the Model

**Either manually or using a python package, we can ADD one variable at a time to see the affect it has on the overall model. SequentialFeatureSelector is a python package that adds the next most relevant variable to the model, sequentially. The adjusted R-squared values will increase until a variable is added that no longer helps the model. The previous model (without the last variable added) is now the best fit model to explain variations in the dependent variable .**



```
Features: 43/54 -- score: 0.7535547320272855
```

```
Features: 44/54 -- score: 0.7535547320272965
```

```
Features: 45/54 -- score: 0.7535547320272904
```

SequentialFeatureSelector runs one variable at a time and gives an output of the adjusted R-squared values. As indicated in the partial output above, adding the 44<sup>th</sup> variable increased the p-value slightly, meaning this variable is relevant and adds value to the model. The 45<sup>th</sup> variable, however, drops the adjusted R-squared score slightly. This means that the 45<sup>th</sup> variable did not help the model and should not be added. Therefore, the best model is obtained after adding the 44<sup>th</sup> variable. Python can then output the corresponding 44 variables in list format for analysis.

# Techniques to Evaluate Model

## Mean Absolute Error (MAE)

**MAE** calculate the residual for every data point, taking only the absolute value of each so that negative and positive residuals do not cancel out. The average is then taken which describes the typical magnitude of the residuals We then take the average of all these residuals. Effectively, MAE describes the typical magnitude of the residuals.

## Root Mean Square Error (RMSE)

The root mean square error (RMSE) is similar to the MAE but squares the difference before summing. RMSE can be interpreted as the standard deviation of the unexplained variance. Lower values of RMSE indicate better fit.

## R-squared

The R-squared variance indicates how the independent variables explains the variation of the dependent Price variable.

## Testing the Linear Regression Assumptions

Five assumptions are made about linear regression models: 1) No multicollinearity 2) Means of residuals approximately equal zero 3) No heteroscedacity 4) Linearity of variables 5) Normality of error terms





# 05

## **Recommendations**

## Design Dashboard

## Improve Dataset

## Retool Equation

### Key Action 1

Design a user-friendly dashboard where categorical variables can be selected from drop down list and numerical variables can be typed in.

As sales of used cars continues, Cars4U sales reps need to record as much info about the cars as possible and include in the dataset.

As more data is added, re-examine the variables that do not add value currently.

### Rationale

The equation is cumbersome and takes time to input values and calculate manually.

As more data is entered, Price fluctuations can be more easily explained.

With more data, values that are currently not helpful may turn out to provide key information.

### Key Action 2

Develop a Cars4U app so sales people can use the pricing equation on the go.

Research prices of electric cars or other vehicles not well represented in the dataset

Look to add other variables to increase accuracy. People may be willing to pay more for a red car!

### Rationale

Cars4U can access key information anytime, anywhere

This will allow Cars4U sales reps to apply similar equations to electric cars and other vehicles

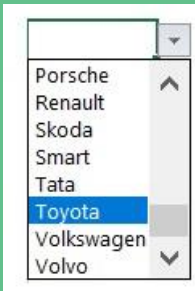
R-squared = 0.762 currently but can go up. As more of the price variations are explained, the better the model will function.

### Suggested Retail Price

**\$ 16,167.06** USD

**₹ 12.44** Lakh

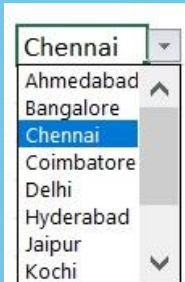
### Brand Name



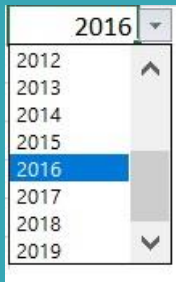
### Kilometers Driven

**24,000**  
km

### Location



### Year



### Power

**100**  
bhp

### Mileage

**12.8**  
kmpl  
or  
km/kg

Name ▾	Location ▾	Year ▾	KM_Driven ▾	Mileage ▾	Power ▾	USD ▾	Lakh ▾
Toyota	Chennai	2016	24000	12.8	100	\$16,167.06	12.44

Behind the dashboard, the equation can be imbedded in a table format. Selecting the Name of the car, in this case Toyota, ‘turns on’ the column Toyota, giving it a 1 value and ‘turns off’ all other Names. This means that  $1 \times (-9,154.71)$  is added to the equation, where  $(-9,154.71)$  is the coefficient of Toyota. All other Name coefficients are multiplied by 0 when ‘turned off’. The same situation occurs for Location: only Chennai is “turned on” so  $1 \times 1,158.98$  is added to the equation. The next four variables (Year, KM\_Driven, Mileage and Power) are always on and the imputed value is simply multiplied by the coefficient: ie. 12.8 (input) x mileage coefficient  $(-83.52)$ . All of the variables  $> 0$  are multiplied by their respective coefficients, added together along with the constant  $(-2,062,000)$  to determine the suggested price of the used vehicle being looked up. For the above search, the following calculation is performed:

$$\text{Price} = -2,062,000 + 1029.59\text{Years} + (-0.0127)\text{KM\_Driven} + (-83.52)\text{Mileage} + 118.74\text{Power} + (-9,154.71)\text{Toyota} + 1,158.98\text{Chennai}$$

$$\text{Price} = -2,062,000 + 1029.59(2016) + (-0.0127)(24,000) + (-83.52)(12.8) + 118.74(100) + (-9,154.71)(1) + 1,158.98(1)$$

$$\text{Price} = \$16,167.06$$



## Revenue

The pricing model is designed to help Car4U sales reps set prices on inventory so they never sell used cars below their market value again.

The model not helps shield against selling cars under value, but it also prevents setting the price of a car too high making Cars4U an uncompetitive option for employees.

## Expenses

Despite being designed to set prices on merchandice, the model can also minimize expenses. Knowing the resale value of the car before Cars4U acquires it for inventory can help the company avoid overpaying for a car that may require the company to sell at a loss to get it off the books.

## Total Profit

Cars4U could actually set profit models based on our profit model. For example, it could dicate that inventory cannot be purchased unless the company can mark it up by xx%. For example, if the price guide suggests a specific model of Toyota should resell for \$15,000 and the profit margin is set to 25%, then the company should not pay more than \$12,000 for the car.



# 06

## **Conclusions**

**After training the computer extensively, the best Price formula is:**

$$\begin{aligned} \text{Price} = & -2,062,000 + 1029.59[\text{Years}] - 0.0127[\text{KM Driven}] - 83.52[\text{Mileage}] + 118.74[\text{Power}] - \\ & 13,560.00[\text{Hyundai}] - 13,640.00[\text{Isuzu}] + 8,037.28[\text{Jaguar}] - 9,864.59[\text{Jeep}] + \\ & 82,440.00[\text{Lamborghini}] + 18,360.00[\text{Land Rover}] - 13,710.00[\text{Mahindra}] - 12,060.00[\text{Maruti}] + \\ & 1,771.70[\text{Mercedes-Benz}] + 5,532.68[\text{Mini}] - 11,010.00[\text{Mitsubishi}] - 13,170.00[\text{Nissan}] + \\ & 13,880.00[\text{Porsche}] - 13,670.00[\text{Renault}] - 13,960.00[\text{Skoda}] - 14,180.00[\text{Tata}] - 9,154.71[\text{Toyota}] - \\ & 13,670.00[\text{Volkswagen}] - 7,272.09[\text{Volvo}] + 1,952.92[\text{Bangalore}] + 1,158.98[\text{Chennai}] + \\ & 1,793.26[\text{Coimbatore}] - 1,056.68[\text{Delhi}] + 1,997.84[\text{Hyderabad}] + 1,168.54[\text{Jaipur}] - 574.31[\text{Kochi}] - \\ & 1,286.16[\text{Kolkata}] - 1,227.45[\text{Mumbai}] + 414.56[\text{Pune}] \end{aligned}$$

**PRICE FORMULA**

Simplifying the above formula based on the current data available will result in lowering the percent of variations in the Price data below 75%. Removing Name, which would simplify the model by removing 30+ variables, would drop its predictive power by almost 7.5%.



# Techniques to Evaluate Model

## Mean Absolute Error (MAE)

Using the Adding Variables Method, we were able to improve MAE from 5273.09 to 3850.10. This is not an amazing result, since it means, on average, the predicted price is about \$3850 plus or minus of what the actual price is.

## Root Mean Square Error (RMSE)

Using the Adding Variables Method, we were able to improve from 8699.87 to 6938.75. There is still quite a bit of standard deviation around the unexplained price variance, but it does prove we are improving the model with each added variable.

## R-squared

Using the Adding Variables Method, we improved the R-squared from 0.6033 to 0.762. This is a big improvement, as the added variables help explain an additional 16% of the price variance

# Did the Model Improve?

	First Manual Trial	Final Manual Trial	Machine Learning Test	Machine Learning Training
MAE	5273.09	3850.1		
RMSE	8699.87	6938.75	6296.77	6880.93
R-squared	0.6033	0.747	0.762	0.760

The above table shows the substantial improvements as the model progressed to its final form. When Price was explained using just one variable (Power), it only explained 60% of the variations in the Price data and it had high errors. By continuously adding variables until R-squared no longer improved, we were able to raise the R-squared value enough to explain almost 75% of the Price variations. Using Machine Learning, we could improve the model even further, explaining 76% of the Price variations and decreasing the RMSE even further.

# Checking Linear Regression Assumptions

## 1. No Multicollinearity

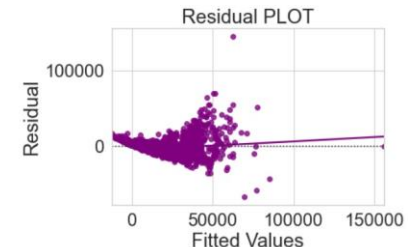
We tested for Multicollinearity by using Variance Inflation Factors (vifs). Variables with scores above 10 were identified and the highest scoring variable was dropped from our model. We repeated this process until the model was free of Multicollinearity. Engine and Fuel Type variables were dropped during this process.

## 2. Mean of residuals should be 0

Using python, we found that the mean of the residuals of our final model is  $-6.18e-09$ . This is a very small value that is very close to 0.

## 3. Test for Linearity

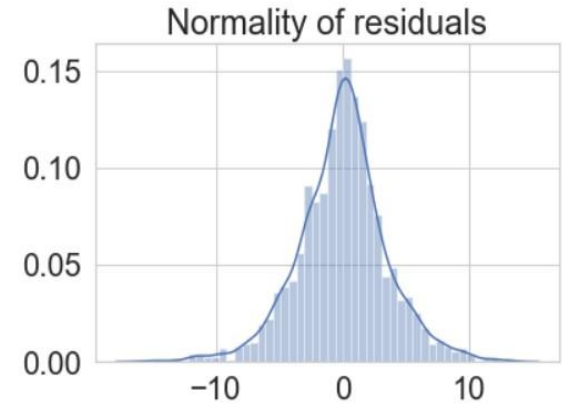
Plotting the fitted values vs Residuals should show no pattern. If true, the model is linear otherwise model is showing signs of non-linearity. The graph to the right does not show a discernable pattern so we can conclude the assumption is satisfied



# Checking Linear Regression Assumptions

## 4. Test for Normality

**The error terms/Residuals should be normally distributed. We can check by graphing them. The graph to the right shows the Residuals are near-normal. We can conclude that this assumption has been met.**



## 5. Test for Homoscedasticity

**Homoscedasticity is achieved when the variance of the residuals are symmetrically distributed across the regression line. After performing a goldfeldquandt test on the residuals, if p-value < 0.05, homoscedasticity is achieved. The output on the right shows this assumption has been met.**

`('p-value', 3.86167090606552e-14)]`

**After training the computer extensively, the best Price formula is:**

$$\begin{aligned} \text{Price} = & -2,062,000 + 1029.59[\text{Years}] - 0.0127[\text{KM Driven}] - 83.52[\text{Mileage}] + 118.74[\text{Power}] - \\ & 13,560.00[\text{Hyundai}] - 13,640.00[\text{Isuzu}] + 8,037.28[\text{Jaguar}] - 9,864.59[\text{Jeep}] + \\ & 82,440.00[\text{Lamborghini}] + 18,360.00[\text{Land Rover}] - 13,710.00[\text{Mahindra}] - 12,060.00[\text{Maruti}] + \\ & 1,771.70[\text{Mercedes-Benz}] + 5,532.68[\text{Mini}] - 11,010.00[\text{Mitsubishi}] - 13,170.00[\text{Nissan}] + \\ & 13,880.00[\text{Porsche}] - 13,670.00[\text{Renault}] - 13,960.00[\text{Skoda}] - 14,180.00[\text{Tata}] - 9,154.71[\text{Toyota}] - \\ & 13,670.00[\text{Volkswagen}] - 7,272.09[\text{Volvo}] + 1,952.92[\text{Bangalore}] + 1,158.98[\text{Chennai}] + \\ & 1,793.26[\text{Coimbatore}] - 1,056.68[\text{Delhi}] + 1,997.84[\text{Hyderabad}] + 1,168.54[\text{Jaipur}] - 574.31[\text{Kochi}] - \\ & 1,286.16[\text{Kolkata}] - 1,227.45[\text{Mumbai}] + 414.56[\text{Pune}] \end{aligned}$$

**PRICE FORMULA**

Simplifying the above formula based on the current data available will result in lowering the percent of variations in the Price data below 75%. Removing Name, which would simplify the model by removing 30+ variables, would drop its predictive power by almost 7.5%.

## Simplifying Formula to 4 variables:

$$\text{Price} = -2,061,380.45 + 197.89[\text{Power}] + 1019.34[\text{Year}] - 11.28[\text{Mileage}] - 0.158[\text{KM Driven}]$$

Simplifying the equation looks beneficial, but it drops the R-squared value to 66.6%. As a result, solving for the 2016 Toyota, with Power = 100, Mileage = 10.8 and KM Driven = 24,000 returns the value of \$12,895.93, which is \$3,271.13 below what our model predicted. This is highly problematic and shows the cost associated with over-simplifying the model.

**Key Finding:** As the complexity in the model



the accuracy



## Simplifying Formula to Customer-known variables:

$$\text{Price} = -2,061,380.45 + 197.89[\text{Power}] + 1019.34[\text{Year}] - 11.28[\text{Mileage}] - 0.158[\text{KM Driven}]$$

Intercept of the linear equation: [-2089022.88301591]

Coefficients of the equation are: [[ 1.05361357e+03 -9.35906756e-03 4.09527420e+02 4.98475936e+04  
 -2.62062720e+04 -3.18183863e+04 -2.58268444e+04 -1.97891903e+04  
 -2.33731368e+04 6.67641871e-08 -2.46146821e+04 -2.54728428e+04  
 -1.58890397e+04 1.00514469e+04 -1.22761196e+04 1.26897133e+05  
 2.23103210e+04 -2.24428637e+04 -2.63224967e+04 1.68573837e+03  
 2.82142555e+02 -1.68053125e+04 -2.58216595e+04 4.25279723e-09  
 3.49739986e+04 -2.60502361e+04 -2.14660101e+04 -1.23691279e-10  
 -2.78751121e+04 -1.65898898e+04 -2.54455334e+04 -7.50016151e+03  
 1.91859194e+03 1.49416222e+03 1.93702405e+03 -6.30299133e+02  
 1.36836806e+03 5.47081786e+02 -5.86366493e+02 -1.67785790e+03  
 -7.90766796e+02 2.00439494e+01]]

It can be argued that customers would not necessarily ask for used vehicles by mileage or power, preferring basic statistics like name, location, year and mileage. However, like the previous attempt at simplifying the model, the R-squared is also reduced, this time to 65.3%. Again, applying the formula to the 2016 Toyota from Chennai with 24,000km, we get: 19,741.73. Using this simplified model our price is now \$3,574.67 higher than using our model. This may sound good, but few customers will be willing to overpay for a used car and will probably purchase a new vehicle instead.

Again, this is highly problematic and shows the cost associated with over-simplifying the model.

## CONSUMER BEHAVIOUR

- 1 Customers will pay more for a powerful car.  

---
- 2 Brand recognition is strong. The type of brand plays a big role in how prices are determined.  

---
- 3 Customers will pay more for newer vehicles. Price drops as the car ages.  

---
- 4 The city where the car is purchased helps determine the price. Hyderabad and Bangalore command higher prices.  

---
- 5 Customers will pay more for a car with low kilometers. Prices drop as kilometers increase.  

---
- 6 Customers are not concerned about fuel efficiency and will not pay more to own a fuel-efficient car.

## MODEL BEHAVIOUR

- 1 Will only work when all variables are known.  

---
- 2 Equation will fail if attempting to apply to a Brand or Location outside the drop list.  

---
- 3 This model cannot predict the price of electric vehicles, since this variable had no known price to begin with and was expunged from the dataset.  

---
- 4 Attempts to simplify the model to address the above issues will reduce the accuracy of the model. It becomes a trade-off scenario.  

---
- 5 Equation is only relevant for the current year. As time progresses, prices for the models currently in the dataset should go down.



# THANKS

Does anyone have any questions?

mullin.scott@gmail.com  
+91 620 421 838



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.

