Project Information

Project Title: Predicting the prices of pre-owned cars.

Project Motivation:

My father was planning to buy a second hand (pre-owned) car and wanted it to be in a good working condition and in a profitable amount. He was really concerned and worried that if he will get car in a good price or not. On hearing this I thought to use my statistical approach and knowledge to predict the price of a pre-owned car so that my father will not suffer any loss and get a car which fits his requirements and is in a good condition.

Data: The data is of Storm Motors which is an e-commerce company who acts as mediators between parties interested in selling and buying pre-owned cars.

Storm Motors wishes to develop an algorithm to predict the prices of pre-owned cars which satisfies both the seller and the buyer of the car.

The data has been recorded about the seller and car including-

* Specification details (gearbox, power, fueltype)
* Condition of the car (notRepairedDamaged, kilometre)
* Seller details (seller, postalcode)
* Registration details (yearOfRegistration, monthOfRegistration)
* Model information (brand, model, vehicletype)
* Price(price)
* Advertisement details (dateCrawled, name, abtest, dataCreated, lastseen, offerType)

DESCRIPTION OF THE VARIABLES:

* dateCrawled: date when the ad first crawled, all field values are taken from this date
* name : name of the car, brand, model, etc.
* seller : nature of seller ­­­-­­- whether the seller of the car is ‘private’ or ‘commercial’
* offerType : whether the car is on offer or has the buyer requested for an offer
* abtest : two versions of ad – test or control
* vehicleType : types of cars
* yearOfRegistration : Year in which the car was first registered
* gearbox : The type of transmission used by the car -- Automatic / Manual
* powerPS : Power of the car in HP
* model : The model of the car
* kilometre : The total kilometres driven in the car by the previous owner(s)
* monthOfRegistration : Month in which the car was first registered
* fuelType : The type of fuel used by the car
* brand : The brand of the car.
* notRepairedDamage : status of repair for damages – if yes damages have not been rectified; if no damages were taken care of
* dateCreated : date at which the ad at storm motor was created
* postalCode : a code of letters and digits used as part of a postal address; postal code of the seller
* lastSeen : when the crawler saw this as last online

TARGET VARIABLE:

* price : Continuous target variable which tells the price on the ad to sell the car (in $)

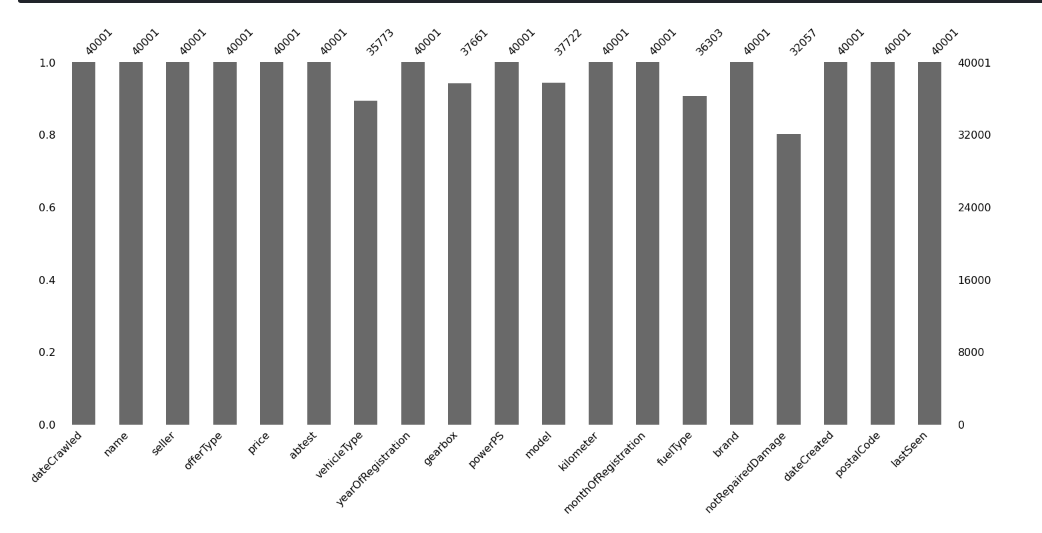
**Steps to proceed with data cleaning:**

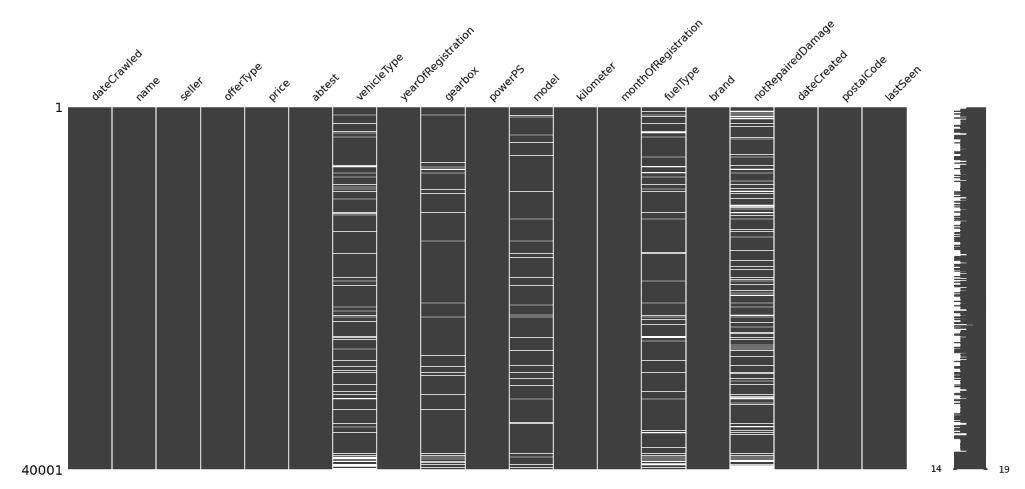
1. Split the data into train and test set. Test set should not contain any of the missing values of response variable i.e. ‘price’.
2. Keep this test set aside.
3. The most important thing before using any technique for imputation is checking the ‘Missing pattern’ in the data. If the missing data is completely at random, then using techniques for imputation will be a good choice to follow.
4. Check for the missing data in independent attributes and find a suitable technique henceforth.
5. Now, take the train set and further make 3 buckets – 1 for train (with no missing value), 1 for validation (with no missing data) and the last one should be the set containing just missing values of ‘price’.
6. Imputing of irrelevant values of price using various methods.
7. Use an appropriate technique to predict the missing values of price.

Analysing the pattern of Missingness

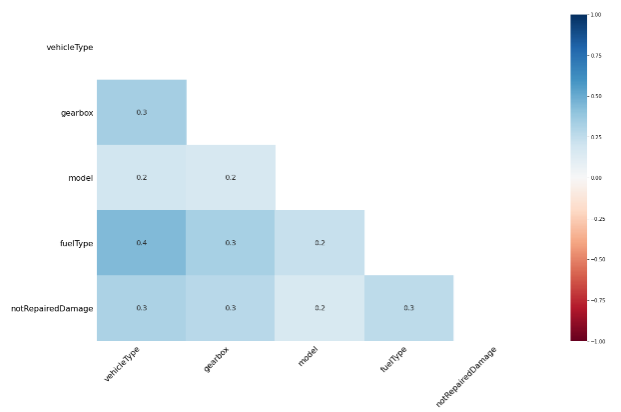
Missing Columns: VehicleType, gearbox, model, fuelType, and notRepairedDamage

Number of missing values in each independent variable





CORRELATION BETWEEN MISSING VARIABLES:



The frequency (in percentage) for different categories of different Categorical Variables are :

matiz 0.001150

laguna 0.003925

c\_klasse 0.023724

freelander 0.000750

yaris 0.002825

v70 0.001700

delta 0.000150

corolla 0.001625

x\_trail 0.000725

galant 0.000550

**Name: model, Length: 248, dtype: float64**

request 0.000075

offer 0.999925

**Name: offerType, dtype: float64**

manual 0.733332

automatic 0.208170

missing\_gear 0.058499

**Name: gearbox, dtype: float64**

control 0.484313

test 0.515687

**Name: abtest, dtype: float64**

missing\_fuel 0.092448

petrol 0.604510

cng 0.001525

other 0.000575

hybrid 0.000725

lpg 0.016025

electro 0.000225

diesel 0.283968

**Name: fuelType, dtype: float64**

13/03/2016 2:37 0.000025

29/03/2016 12:54 0.000100

7/3/2016 12:51 0.000100

4/4/2016 8:37 0.000050

16/03/2016 21:39 0.000150

1/4/2016 9:55 0.000025

9/3/2016 16:43 0.000025

9/3/2016 16:44 0.000100

5/3/2016 22:47 0.000050

22/03/2016 9:54 0.000125

**name: dateCrawled, Length: 11543, dtype: float64**

no 0.702757

yes 0.098648

missing\_damage 0.198595

**Name: notRepairedDamage, dtype: float64**

29/02/2016 0:00 0.000100

17/03/2016 0:00 0.029549

12/11/2015 0:00 0.000025

28/03/2016 0:00 0.034449

15/03/2016 0:00 0.034124

8/3/2016 0:00 0.033874

11/3/2016 0:00 0.032924

27/03/2016 0:00 0.029249

26/03/2016 0:00 0.032049

27/02/2016 0:00 0.000100

**Name: dateCreated, Length: 65, dtype: float64**

commercial 0.00005

private 0.99995

**Name: seller, dtype: float64**

12/3/2016 10:45 0.000050

1/4/2016 3:41 0.000050

29/03/2016 12:54 0.000025

19/03/2016 6:38 0.000025

16/03/2016 21:39 0.000025

19/03/2016 16:19 0.000125

24/03/2016 6:44 0.000025

9/3/2016 16:44 0.000100

31/03/2016 1:42 0.000025

17/03/2016 3:44 0.000075

**Name: lastSeen, Length: 9507, dtype: float64**

missing\_vehicle 0.105697

suv 0.040674

small car 0.214495

others 0.008775

bus 0.079548

coupe 0.051549

cabrio 0.061723

limousine 0.258819

station wagon 0.178721

**Name: vehicleType, dtype: float64**

Rover\_214i 0.000025

Ford\_focus\_tuev\_bis\_2018 0.000025

Peugeot\_306\_Cabriolet\_1.6\_Pininfarina 0.000025

Peugeot\_307\_75\_Presence\_1.\_Hand 0.000025

Mercedes\_Benz\_\_A\_Klasse\_W140 0.000025

Saab\_9\_3\_1.8\_t\_Arc\_+\_BRC\_Autogasanlage/LPG\_+\_AHK 0.000025

Mitsubishi\_i\_MiEV 0.000025

Renault\_Twingo\_EZ\_2000\_Leder 0.000025

BMW\_530d\_M\_Paket\_Headup\_Keyles\_Go\_TÜV\_2018 0.000025

Peugeot\_306\_Break\_1\_6i 0.000025

**Name: name, Length: 31693, dtype: float64**

**Percentile for price**

np.percentile(data2['price'],95)

Out[30]: 19500.0

np.percentile(data2['price'],99)

Out[31]: 36900.0

np.percentile(data2['price'],98)

Out[32]: 27990.0

np.percentile(data2['price'],98.5)

Out[33]: 31500.0

np.percentile(data2['price'],25)

Out[34]: 1100.0

np.percentile(data2['price'],10)

Out[35]: 450.0

np.percentile(data2['price'],1)

Out[36]: 0.0

np.percentile(data2['price'],5)

Out[37]: 150.0

np.percentile(data2['price'],3)

Out[38]: 0.0

np.percentile(data2['price'],4)

Out[39]: 1.0

np.percentile(data2['price'],4.5)

Out[40]: 100.0

**Percentile capping for Kilometer**

At 5% , the values of the kilometres capped.