

Dataset description:

Data was collected by web crawler from Polish Car trading website - Otomoto.pl

It consists of around 19708 observations (some of them are duplicated) and 18 variables

Objective:

Main objective of this analysis is to select Unsupervised Learning method which will increase fit of Supervised Regression model. Which in this case is Random Forest Regressor.

To perform clustering, 3 models were chosen: K-Means, Mean Shift and DBSCAN.

Variables description:

Price-Price of a car

Brand-Brand of a car

Model-Model of a car

Year_produced-year in which car was produced

Mileage-car's mileage

Cylinders_capacity - Car's cylinders capacity

Fuel_type- Diesel/Petrol etc.

HP - Horse Power

transmission - type of transmission

drive_type - type of drive (FWD,AWD,RWD)

Colour-car's colour

Serviced - whether it was serviced in authorized mechanic

New/Used-Describes whether car was used before or is it brand new

```
data_all.columns
[386] ✓ 0.2s
... Index(['Price', 'Brand', 'Model', 'Year_produced', 'mileage',
        'Cylinders_capacity', 'Fuel_type', 'HP', 'transmission', 'drive_type',
        'liters_per_km', 'Type', 'CO2 emission', 'No_of_doors', 'No_of_seats',
        'Colour', 'Serviced', 'New/Used'],
        dtype='object')
```

Feature engineering:

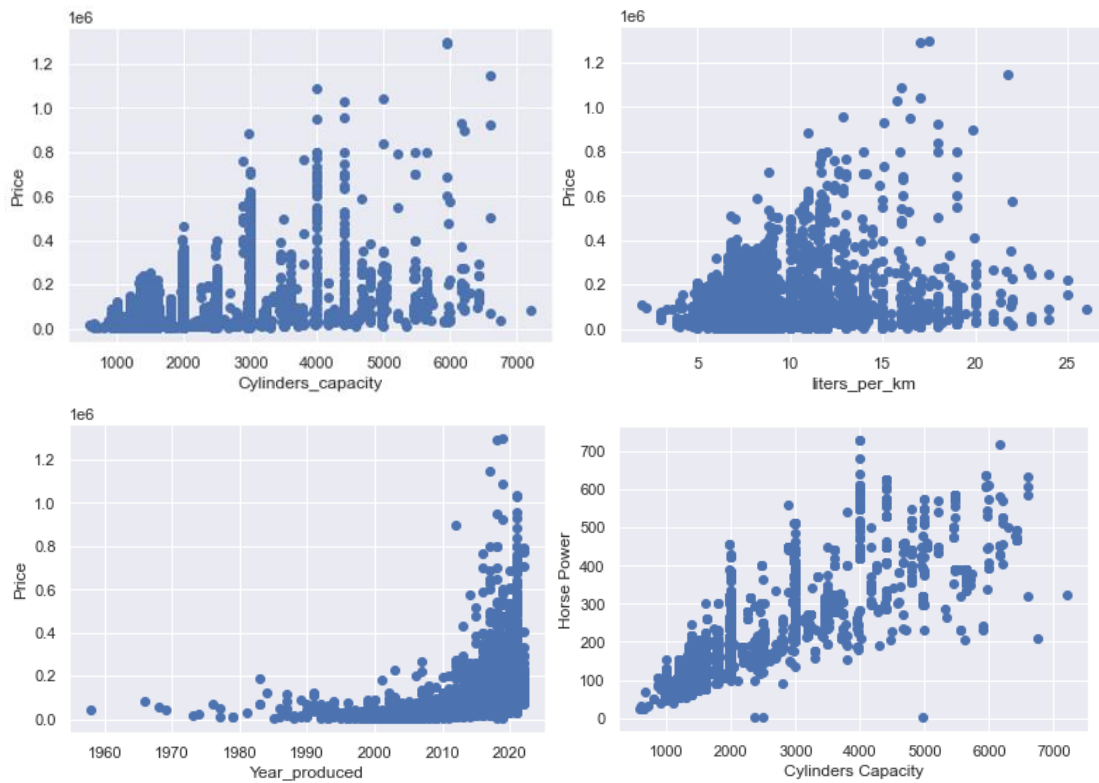
- Removed two outliers (first one because its price was too big and second one because its liters per km ratio was too high)
- Reviewed each variable separately and converted it to an expected type and format
- Filled missing values with mean values of a sub groups
- Removed the rest if there were still missing values
- Grouped car brands and models as 'Other' when they appearing too rarely
- All numeric variables were transformed using log1p transformation due to their skewness

Final 'clean' dataset consists of 8503 observations

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8503 entries, 1 to 19706
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Price                 8503 non-null  float64
1   Brand                 8503 non-null  object
2   Model                 8503 non-null  object
3   Year_produced         8503 non-null  int32
4   mileage               8503 non-null  float64
5   Cylinders_capacity    8503 non-null  float64
6   Fuel_type             8503 non-null  object
7   HP                   8503 non-null  float64
8   transmission          8503 non-null  object
9   drive_type            8503 non-null  object
10  liters_per_km         8503 non-null  float64
11  Type                  8503 non-null  object
12  CO2 emission          8503 non-null  float64
13  No_of_doors           8503 non-null  float64
14  No_of_seats           8503 non-null  float64
15  Colour                8503 non-null  object
16  Serviced              8503 non-null  object
17  New/Used              8503 non-null  object
dtypes: float64(8), int32(1), object(9)
memory usage: 1.2+ MB
```

EDA :

Below few of correlation graphs are presented:



All correlations are as we should expect:

- The higher cylinder capacity the higher the price
- The higher the cylinder capacity the more horse power car has
- The higher the fuel burning rate the higher the price
- The newer the car the more expensive it is

Correlation between variables:

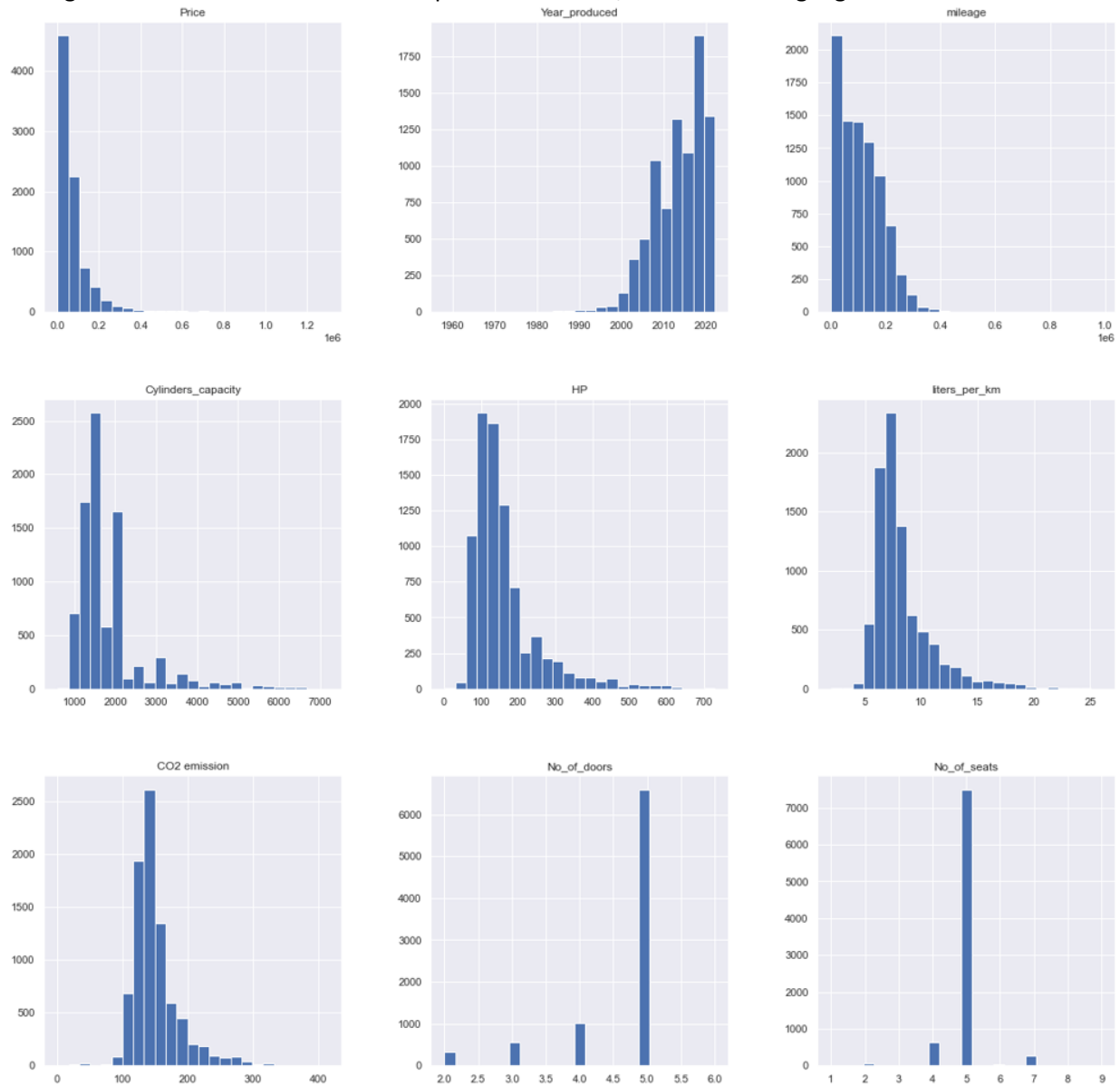
When it comes to direction in which different parameters are correlated with the price:

There are top 30 negatively and positively correlated:

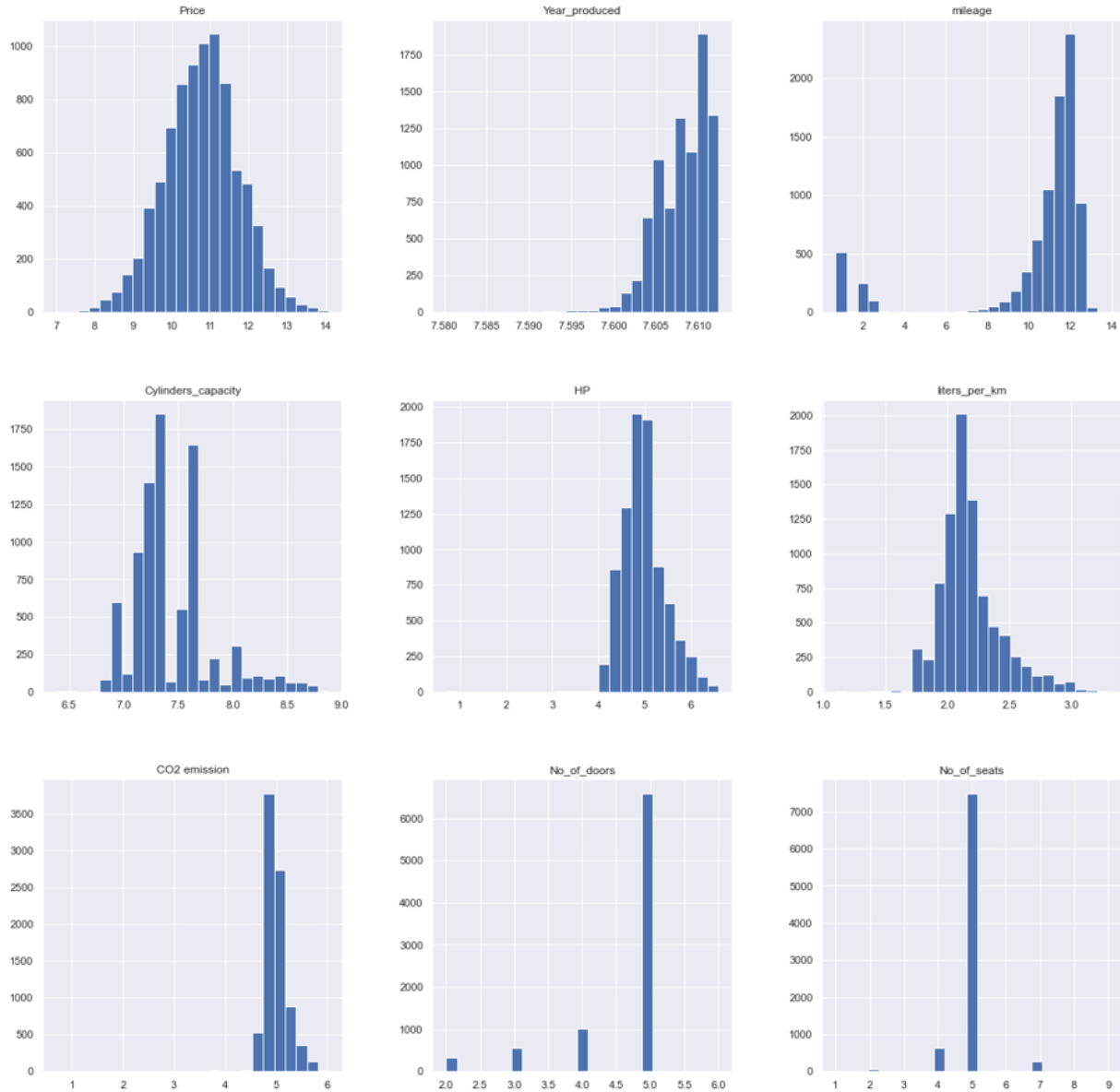
mileage	-0.565052	Price	0.743590
liters_per_km	-0.350799	No_of_doors_5.0	0.220582
transmission_Manual	-0.255964	Colour_White	0.168758
drive_type_RWD	-0.235059	HP	0.150132
CO2_emission	-0.209621	No_of_seats_5.0	0.118961
No_of_doors_3.0	-0.180385	Colour_Gray	0.084075
Fuel_type_Petrol	-0.169636	Model_tipo	0.080530
Cylinders_capacity	-0.168949	Brand_skoda	0.079569
Colour_Silver	-0.134902	Model_q3	0.079486
Model_Other	-0.133237	Brand_dacia	0.075577
Model_sl	-0.105919	Model_arteon	0.072460
Model_a4	-0.096026	Model_xc40	0.071079
Model_seria3	-0.094904	Brand_hyundai	0.063899
No_of_seats_4.0	-0.090991	drive_type_FWD	0.063314
Model_vectra	-0.083228	Model_kuga	0.059911
Colour_Green	-0.073809	Model_stelvio	0.058353
Colour_Other	-0.069774	Model_duster	0.056153
Model_xj	-0.068230	Model_tucson	0.056000
Model_clk	-0.065800	Brand_volvo	0.053735
Model_grandvitara	-0.065063	Brand_kia	0.053553
Model_meriva	-0.063083	Model_spacestar	0.053518
Colour_Gold	-0.059779	Model_xc60	0.053138
No_of_seats_2.0	-0.059733	Serviced_Yes	0.052882
Brand_bmw	-0.059520	Model_superb	0.052706
Model_9-3	-0.059374	Model_compass	0.051220
Brand_saab	-0.059374	Model_tiguan	0.049479
Brand_daihatsu	-0.059018	Model_giulia	0.049156
Brand_honda	-0.058072	Model_ateca	0.048580
Model_klasas	-0.057067	Model_captur	0.047847
Model_accord	-0.055774		
Name: Year_produced, dtype: float64		Name: Year_produced, dtype: float64	

As we can see all of them are more or less in line with common sense

Histograms of continuous variables are presented below, all are indicating high skewness:



Histograms after performing logarithmic transformation look much closer to normal distribution:

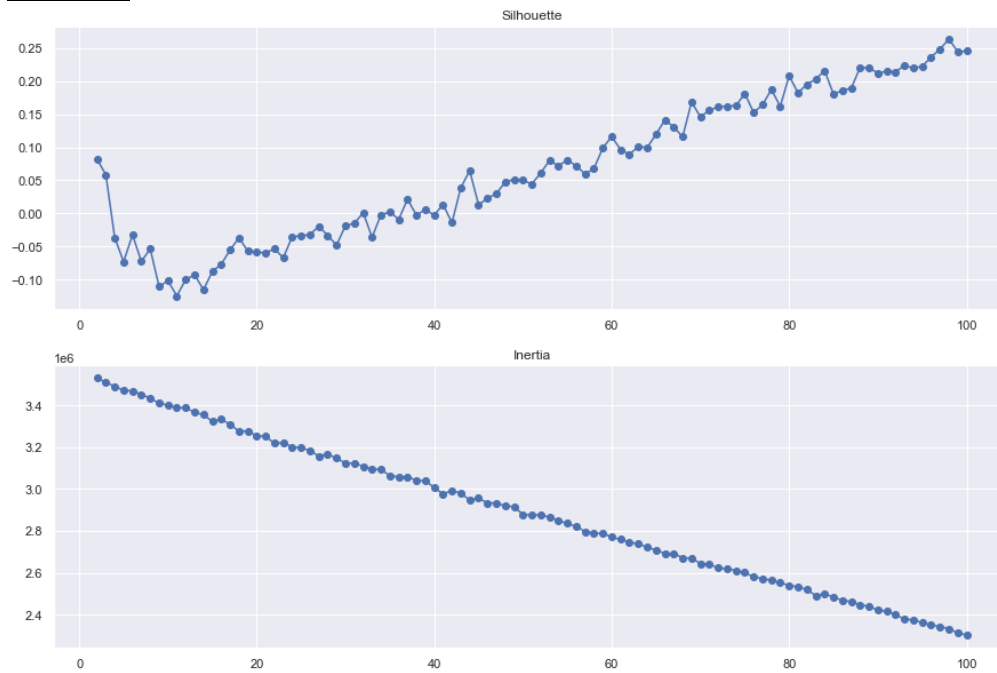


Clustering:

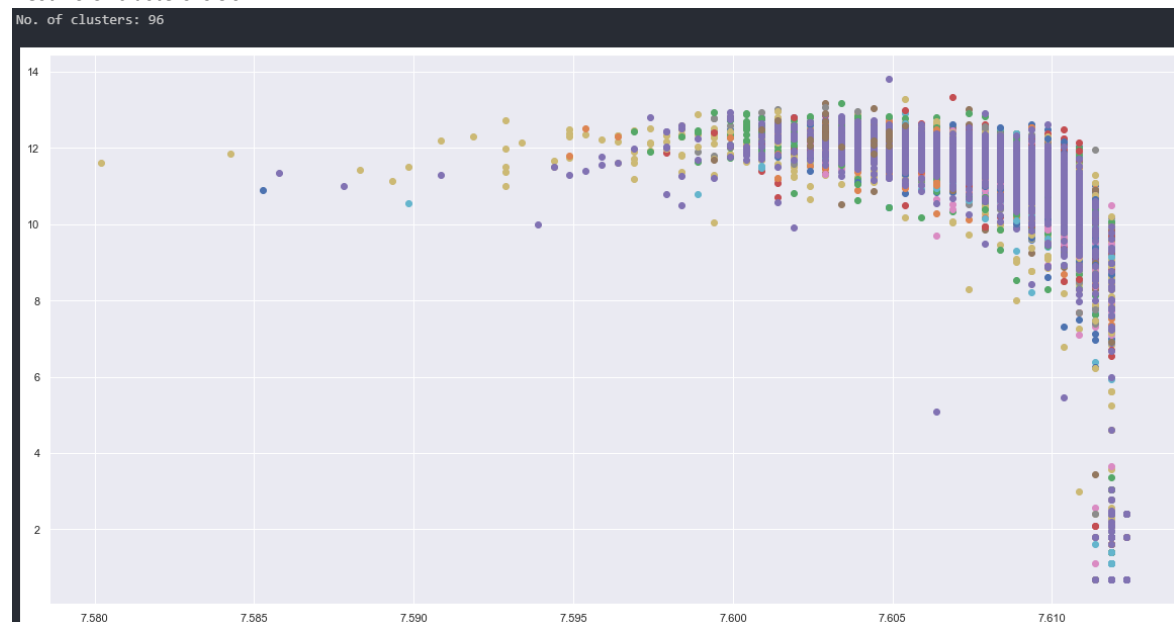
Models used:

- K-Means
- Mean Shift
- DBSCAN

K-Means:

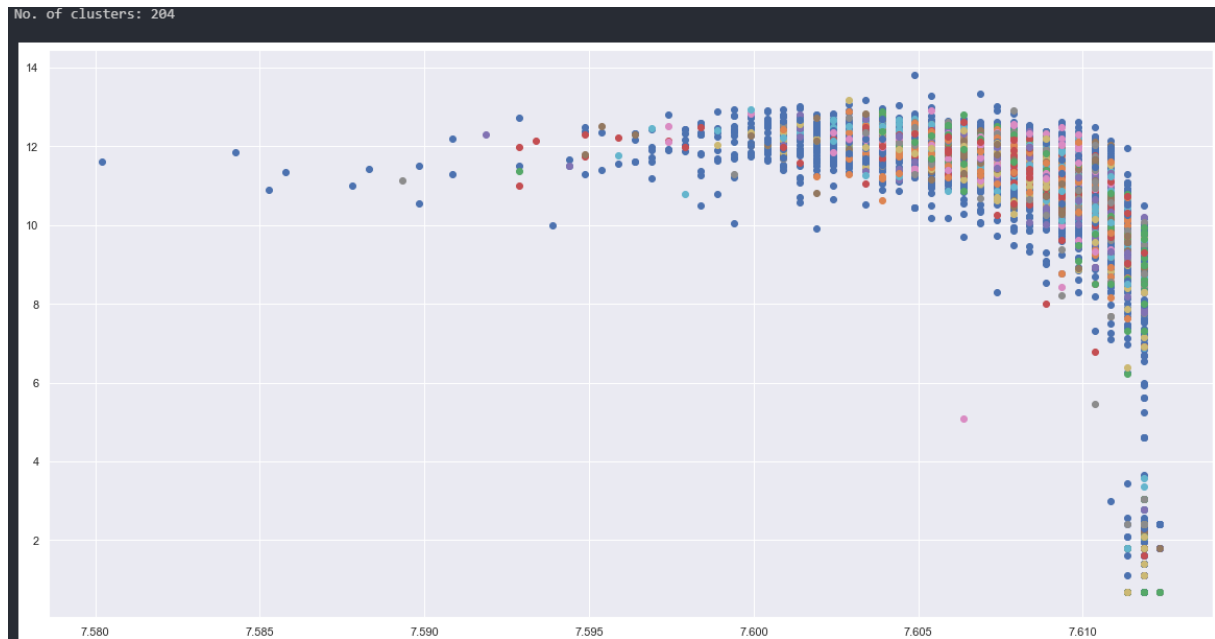


Best no of clusters is 96



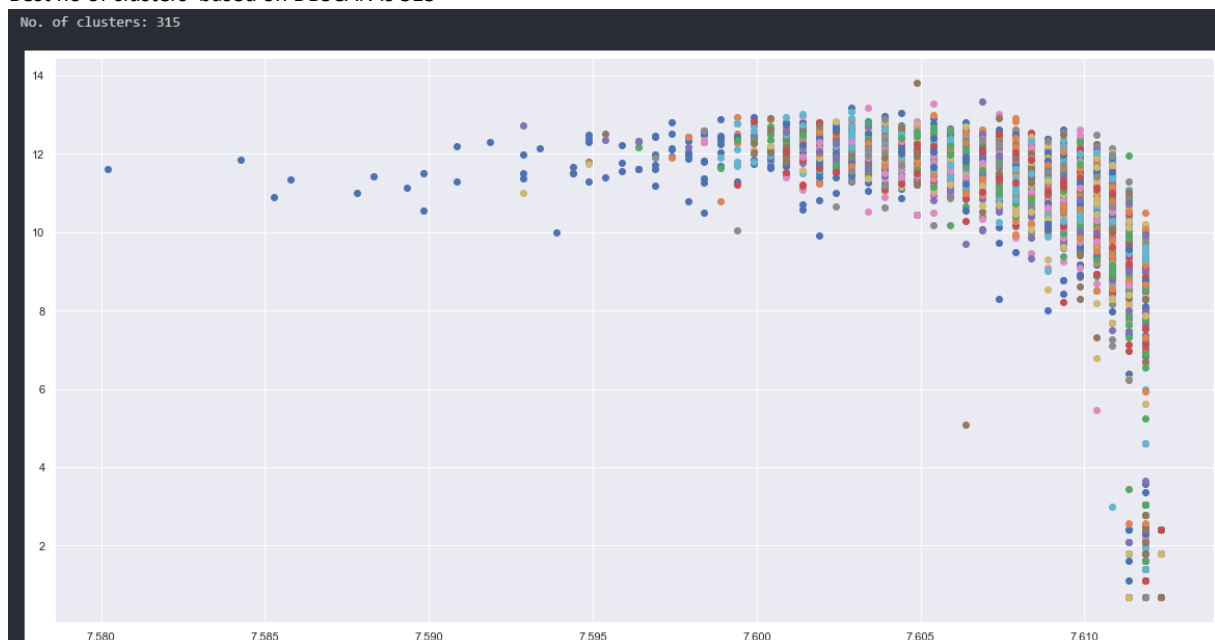
Mean Shift:

Best no of clusters based on Mean Shift is 204



DBSCAN:

Best no of clusters based on DBSCAN is 315

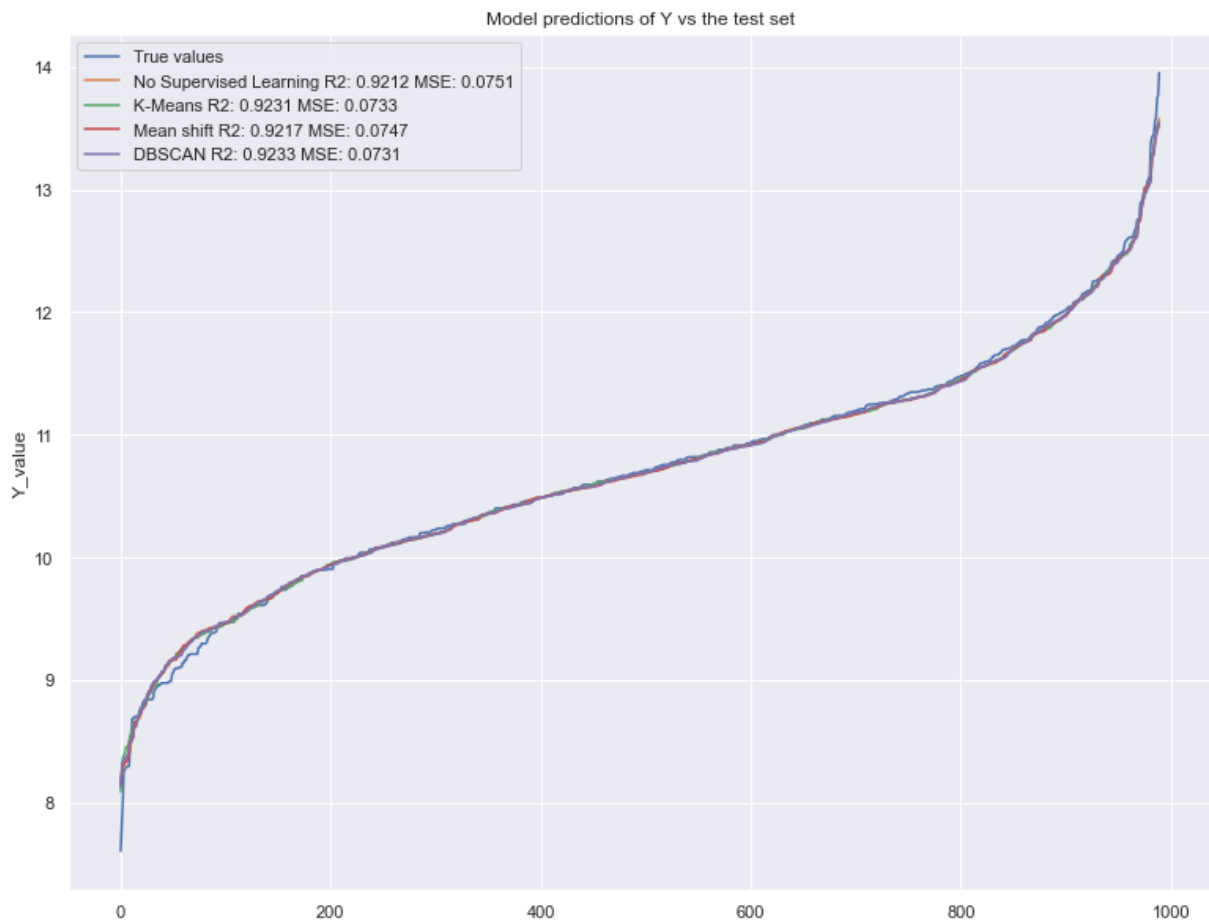


Regression and Summary:

In this part regression was performed based on 4 versions of this dataset

1. Without any Unsupervised Learning clusters
2. With K-Means Clustering
3. With Mean Shift Clustering
4. With DBSCAN clustering

Results are summarized below:



As we can see all in each case clustering increased overall fit to test data, and reduced MSE.

However best results are reached thanks to **DBSCAN** clustering. There is a chance that K-Means, Mean Shift or DBSCAN would bring even better results but it requires a lot of testing of hyperparameters of those clustering methods and then running regression model each time which can be really time consuming.