

1. **PROBLEM DEFINITION:** Predicting the Value of a Used Vehicle

The scope of this project is to predict the price of a used vehicle based on its features.

a. **Client:**

- (1) Individual Buyers: Whoever interested in buying a new car wonders about the actual value of the specific car with the one, which was asked by the seller. So it is important for all to come to a better understanding of the values of the cars.
- (2) Dealers: Most dealers would like to learn the value of that individual car, and determine its value later on.
- (3) Individual Sellers: Most private sellers would need the value of their car since the value is not constant, it changes considering the depreciation, the repairs etc.
- (4) Websites or Applications created to help private parties or dealers sell their vehicles.

b. **Data Set:**

- (1) The data was scraped with from the Ebay. Dataset is acquired from Kaggle: <https://www.kaggle.com/orgesleka/used-cars-database/data>
- (2) Data Set includes 19 features and 371528 data points/observations.
- (3) The data has too many missing values, outliers. Also includes nonsense/ wrong entries like 9999 as year of Registration.
- (4) This data includes the data points/observations from 2016.

2. **DATA WRANGLING:**

a. **Features:**

- (1) Target Feature is 'price'. Since it is the price of an individual car, it is continuous.
- (2) Other features are:
 - dateCrawled : when this ad was first crawled, all field-values are taken from this date
 - name : "name" of the car

- seller : private or dealer
- offerType
- vehicleType
- yearOfRegistration : at which year the car was first registered
- gearbox
- powerPS : power of the car in PS
- model
- kilometer : how many kilometers the car has driven
- monthOfRegistration : at which month the car was first registered
- fuelType
- brand
- notRepairedDamage : if the car has a damage which is not repaired yet
- dateCreated : the date for which the ad at ebay was created
- nrOfPictures : number of pictures in the ad (unfortunately this field contains everywhere a 0 and is thus useless (bug in crawler!))
- postalCode
- lastSeenOnline : when the crawler saw this ad last online

b. Missing Values, outliers, duplicates:

- (1) Duplicates and unnecessary features are deleted. Went over the data points for every feature, found the null values and unique values in each feature with defining a function. This helped me see the number of null values, unique values and unique value counts. I have come to the understanding that some features have only 2 unique values and the percentage of sole value out of 2 unique values is less than 0.1 percent. So this feature could be dropped. Then I decided to drop unnecessary features named ['seller','offerType','nrOfPictures','abtest','monthOfRegistration'].

Column_names	Null_Counts	Unique_Counts	Value_Counts
dateCrawled	0	15623	3/5/16 14:25 68 3/5/16 14:26 62 3/5/16 17:49 58 3/5/16 15:48 58 3/5/16 14:49 55 3/20/16 11:50 55 3/21/16 16:50 55 3/27/16 15:50 55 3/29/16 21:50 55 3/16/16 18:49 55 Name: dateCrawled, dtype: int64
seller	1	3	privat 371534 gewerblich 3 90 1 Name: seller, dtype: int64
offerType	1	3	Angebot 371525 Gesuch 12 golf 1 Name: offerType, dtype: int64
price	1	5597	0.0 10778 500.0 5670 1500.0 5394 1000.0 4649 1200.0 4594 2500.0 4438 600.0 3819 3500.0 3792 800.0 3784 2000.0 3432 Name: price, dtype: int64
abtest	1	3	test 192591 control 178946 4 1 Name: abtest, dtype: int64
vehicleType	37870	9	limousine 95896 kleinwagen 80026 kombi 67564 bus 30202 cabrio 22899 coupe 19016 suv 14708 andere 3357 benzin 1 Name: vehicleType, dtype: int64
yearOfRegistration	1	245	2000 22394 1999 20798 2005 20271 2006 18417 2001 18415 2003 18117 2004 18000 2002 17512 1998 16426 2007 16085 Name: yearOfRegistration, dtype: int64
gearbox	20211	2	manuell 274219 automatik 77109 Name: gearbox, dtype: int64
powerPS	1	1174	0 37244 75 21991 60 14548 150 14033 140 12383 101 12112 90 11577 116 10949 170 10019 105 9503 Name: powerPS, dtype: int64


- Seller feature: 3 out of 371528 observations are dealer. So this feature can be dropped.
- Offer Type feature : 12 out of 371528 observations are Gesuch. So this feature can be dropped.
- Number of Pictures feature has all 0 values. So this feature can also be dropped.
- There are only 29 duplicates that are dropped

(2) Adding features: I decided to add an 'age' column with the help of 'year of registration' feature. This new column is an important aspect regarding the price of the cars (with the following coding)

(3) Cleaned the outliers:

- Number of Cars with newer entries than 2016 : 14680 Number of Cars with older entries than 1970 : 1738. The first of one is the wrong entries since the data had been scraped in 2016. This was filtered. The older cars would not bring any value to the target. And the number of these old cars are 1738. So this also was filtered.
 - Number of Cars with a Value higher than 100.000 : 403 Number of Cars with a Value lower than 250 and equal to 0 : 19998 and 10777. The cars with a high values of over 100.000 would mislead us. These should be handled as outliers. The number of lower value cars are more than I thought. So I will keep them for now.
 - According to my research, the car with higher HP than 600 and lower HP than 5 are wrong entries. So I will handle these outliers by filtering.
- (4) I also changed the non-English words to English. From German: 'nein','ja','benzin','andere','elektro','manuell','automatik','kleinwagen','kombi' , to English: 'no', 'yes', 'gas', 'other', 'electric', 'manual', 'automatic', 'smallCar', 'stationWagon'
- (5) Missing values of the data set are non-numeric features.

dateCrawled	0
seller	1
offerType	1
price	1
abtest	1
vehicleType	37870
yearOfRegistration	1
gearbox	20211
powerPS	1
model	20485
kilometer	1
monthOfRegistration	1
fuelType	33388
brand	2
notRepairedDamage	72062
dateCreated	2
nrOfPictures	2
postalCode	2
lastSeen	2
dtype: int64	



missing_values	
Out[8]: dateCrawled	0
seller	1
offerType	1
price	1
abtest	1
vehicleType	37870
yearOfRegistration	1
gearbox	0
powerPS	1
model	20485
kilometer	1
monthOfRegistration	1
fuelType	0
brand	2
notRepairedDamage	0
dateCreated	2
nrOfPictures	2
postalCode	2
lastSeen	2
dtype: int64	

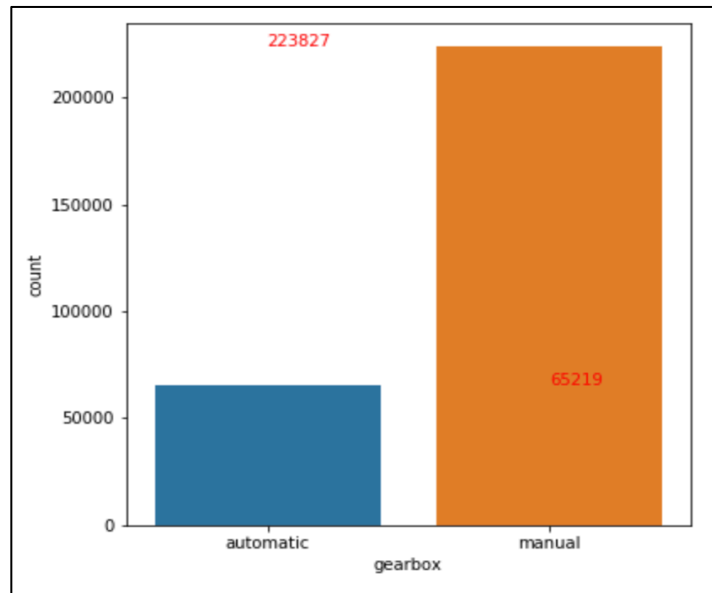
- (6) VehicleType, gearbox, model, fuelType, notRepairedDamage features have missing values. I replaced the Fueltype, gearbox and NotRepairedDmage features' missing values with the most used values. (75% of the vehicles are manuel, 60% of the vehicles are gasoline, 71% of the vehicles are not damaged, missing values of these features are replaced). For vehicle type and model, instead of replacing the missing values with mostly used values, I decided to drop them because they have balanced number of unique values.
- (7) There are some nonsense data and outliers such as yearofRegisttrain: '1000', and '9999', price:0, 100000 (that high price could be a classic car, however it is an outlier since it can not be evaluated among normal used cars) , powerPS:0 or 10000. Filtering these outliers was better before doing the EDA.

```
print('Number of Cars with newer entries than 2016 :',(df['yearOfRegistration'] > 2016).sum())
print('Number of Cars with older entries than 1970 :',(df['yearOfRegistration'] < 1970).sum())
print('Number of Cars more powerful than 600 :',(df['powerPS'] > 600).sum())
print('Number of Cars more expensive than 100000 :',(df['price'] > 100000).sum())
```

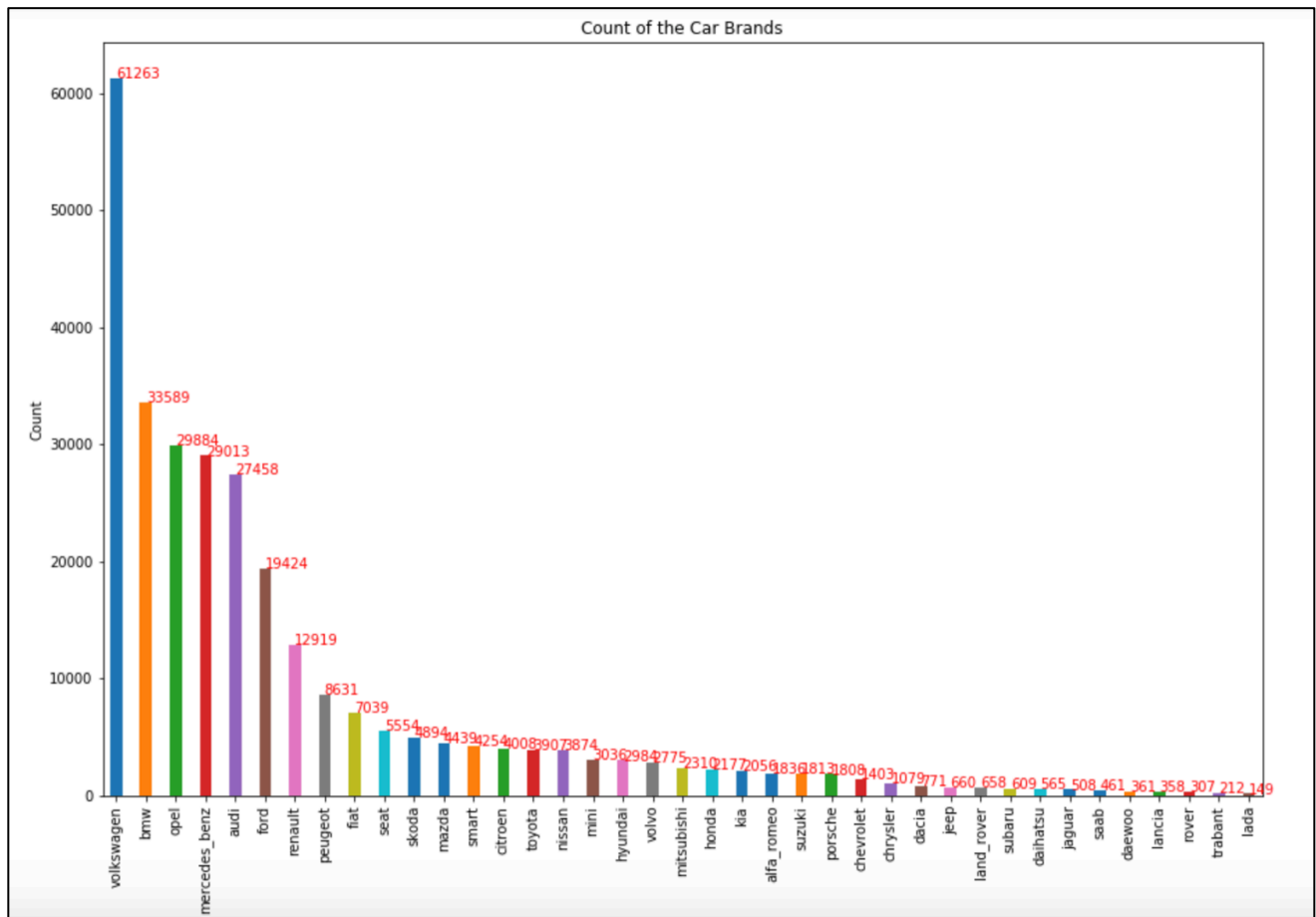
```
Number of Cars with newer entries than 2016 : 19
Number of Cars with older entries than 1970 : 1016
Number of Cars more powerful than 600 : 322
Number of Cars more expensive than 100000 : 271
```

2. DATA VISUALIZATION

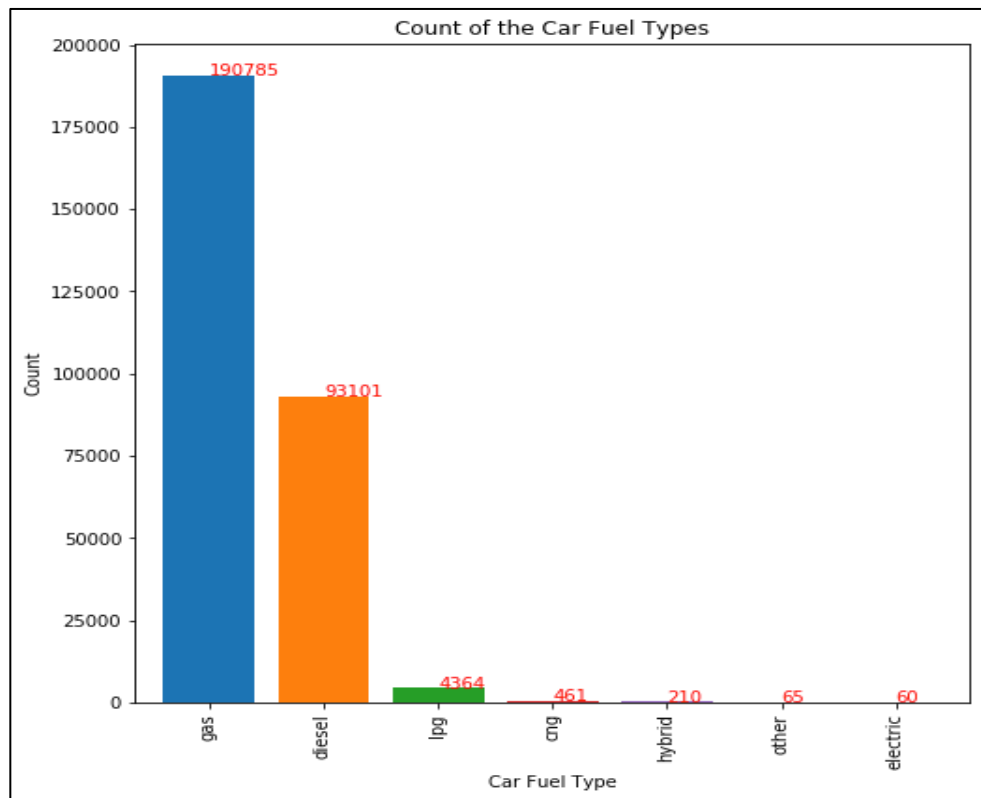
(1) Bivariate Visualisations



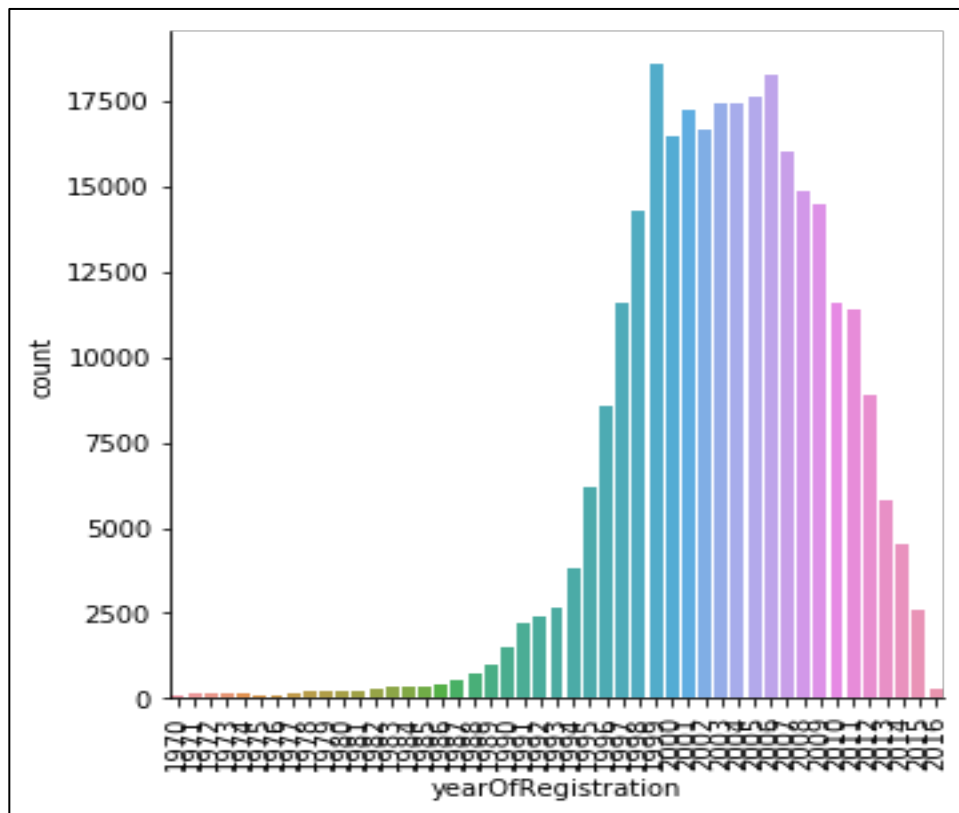
- Regarding the gearbox, manual cars are more widespread



- Regarding the brand, Volkswagen is the most used car, and German cars are more prevalent.



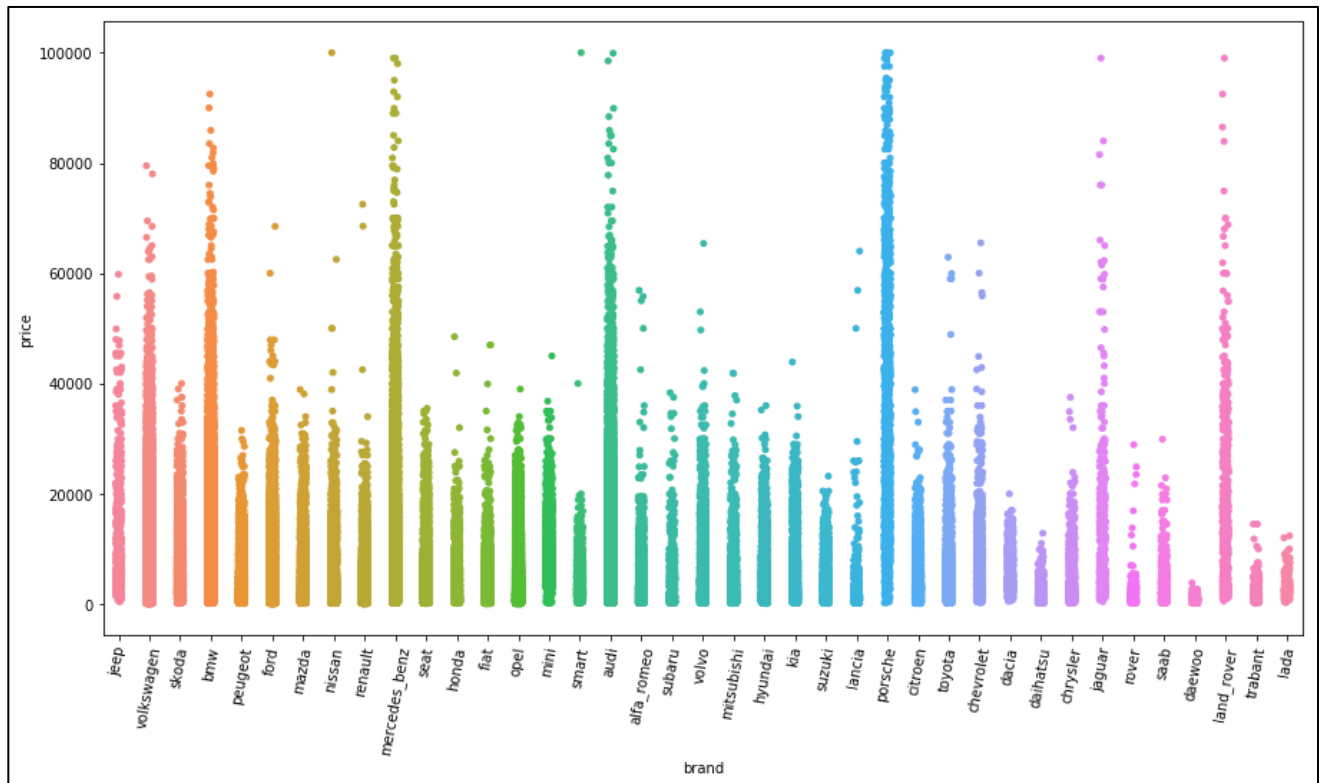
- Since the gas prices in Europe are higher than the US, diesel, lpg and cng cars can be seen. However, regular gas using cars are widespread.



- Most of the used cars are registered between 2000 and 2010

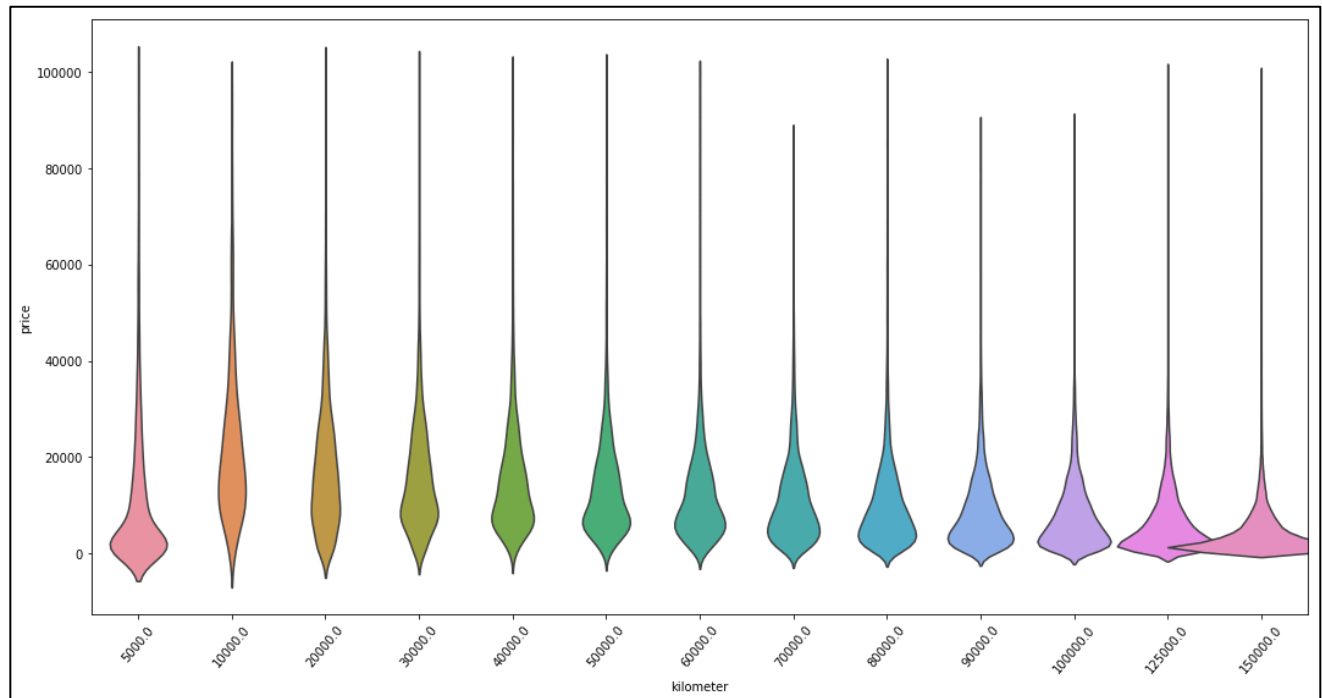
b. Bivariate Visualizations

(1) Price-Brand

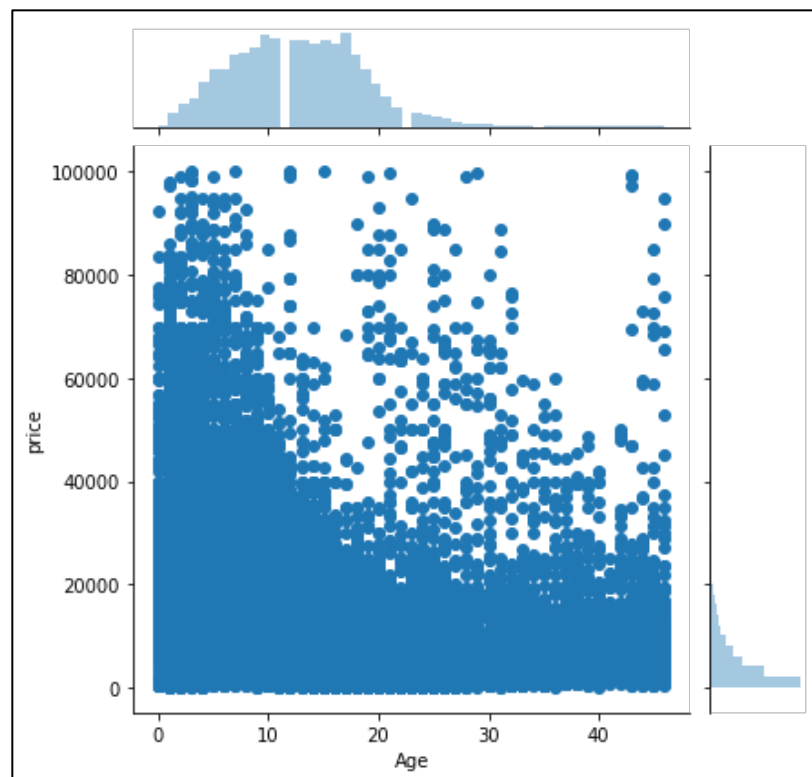


- This chart is a visualization of brand and price features. It is clear that Mercedes_benz, Audi, Porsche, BMW, Land Rover are the most expensive vehicles.

(2) The mileage and the price



(3) Age and the price



- As a general opinion, age is an important factor regarding the price of a car. In this joint plot, we can see that young cars have higher prices than old cars, although there are outliers (very expensive old cars, may be classical or a certain brand)

(4) Correlation Map with multivariate features



- Regarding our target feature 'price', some important related features:
- Age has a negative relationship (-0.52),
- Kilometer has a negative relationship (-0.46)
- PowerPS has a positive relationship (0.52)

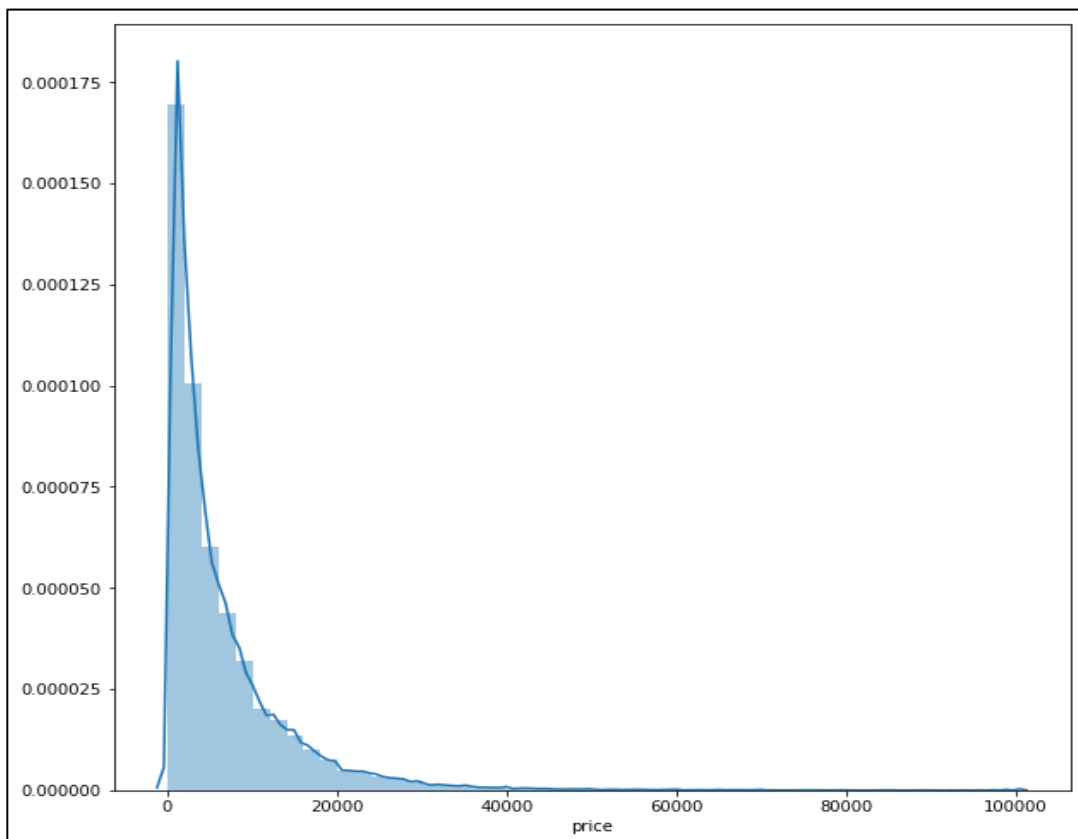
4. INFERENCE STATISTICS

- It is important to find out that distribution of value (price) of the cars normal or not.
- For this, we can perform a hypothesis testing,

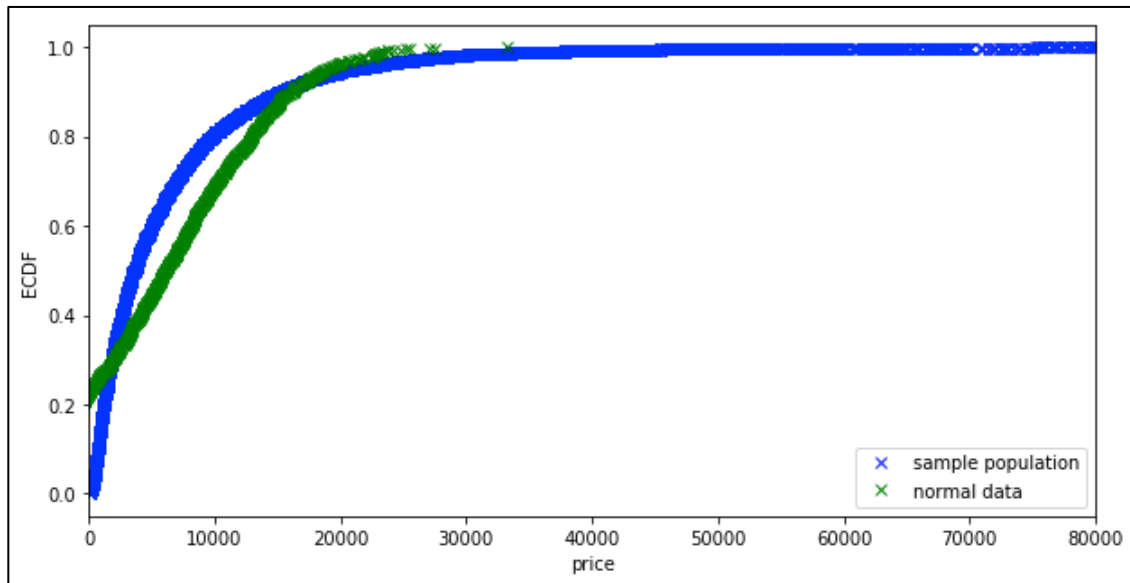
Null Hypothesis: The value of the used cars is a normal distribution

Alternate Hypothesis: The value of the used cars is not a normal distribution

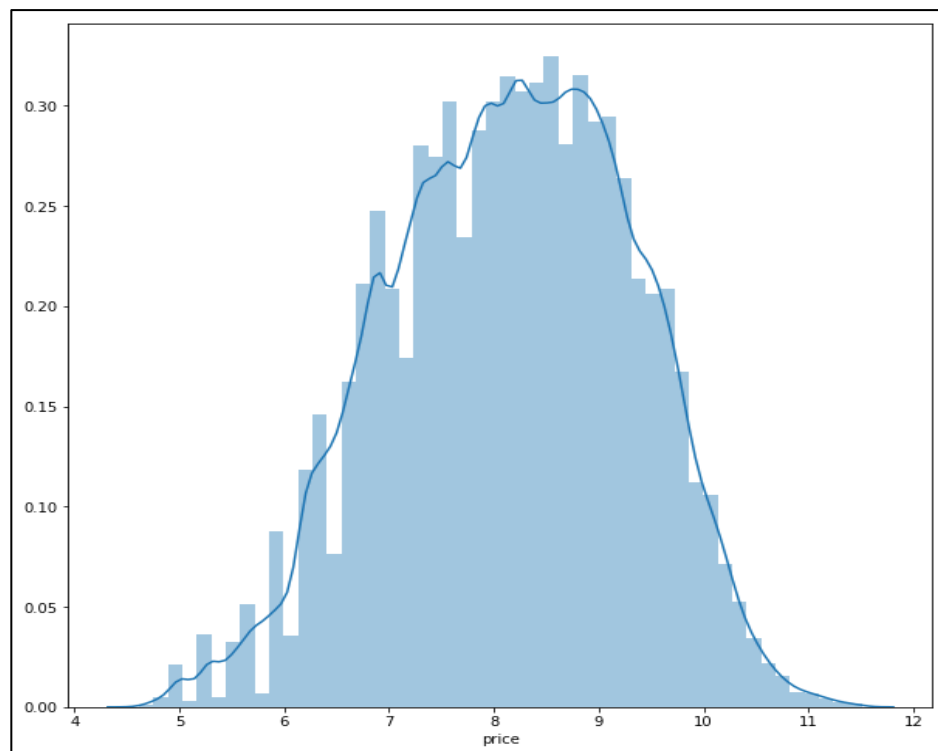
Significance level α is 0.05



- Even though the shape of the plot doesn't look like normally distributed (not like a bell curve), according to central limit theorem, a sample population might be normally distributed in this data set. Thus, we should further select a random sample, test and see the ECDF



- In the normal test, p value is smaller than 0.05 and the shape of ECDF looks like normally distributed, we can reject the null hypothesis. On the other hand, the distribution is right skewed, thus we can convert the price value into a log-price value.



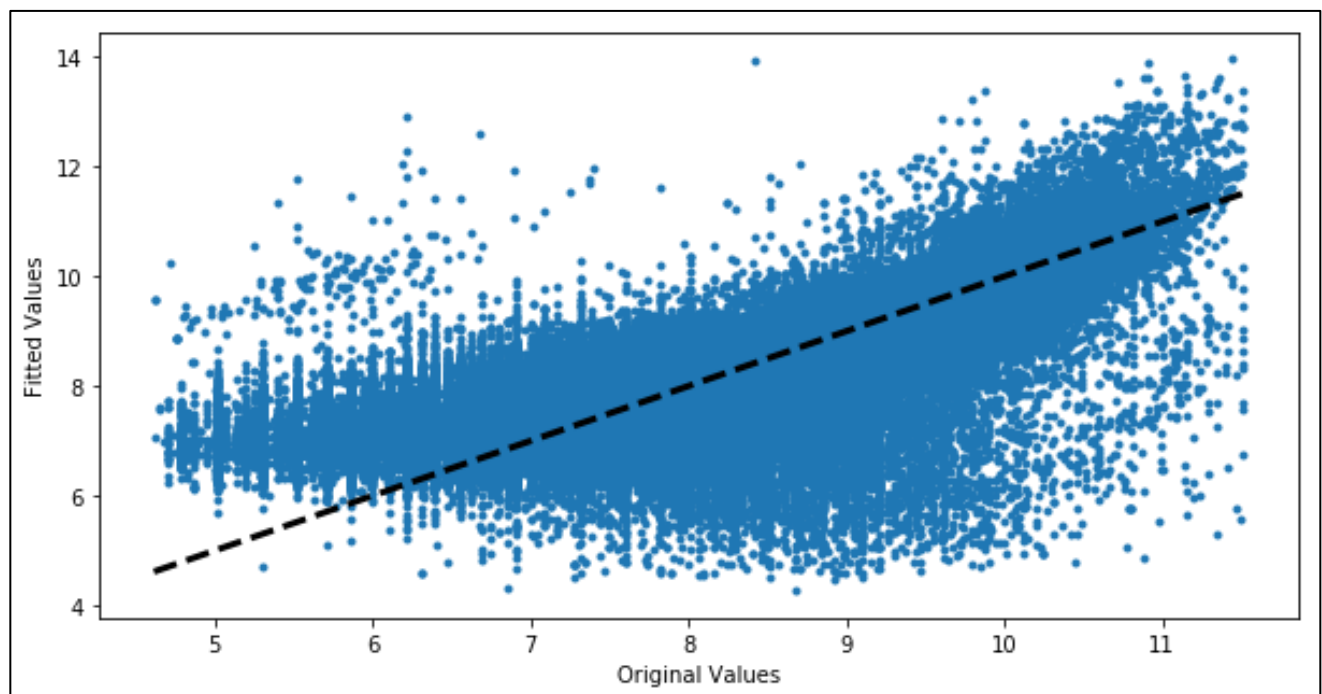
- The log-price distribution now looks normally distributed. So our target feature price is log-normally distributed.

4. MACHINE LEARNING MODELS

This is a supervised learning regression problem since the goal is predicting the price. I first applied a linear regression model to the numeric features in order to see how effective the model works with numeric values. The numeric features are: price, Age , kilometer ,powerPS and yearOfRegistration. I used Scikit Learn and Statsmodel dictionaries. The results of the linear regression with numeric values are not very satisfactory because of the outliers and non-linearity. I also applied Ridge regression, Lasso and Standard Scaling to increase the R square value. The results of statsmodel and the model comparisons are shown below.

a. Result of Linear Regression and Statsmodel (Non-numeric Features):

Similar to the results of our regression model (low R squared(0.65) It seems like our linear model with only numeric features is not very accurate in predicting some values both for lower and higher values as seen in the scatter plot below.



OLS Regression Results						
=====						
Dep. Variable:	price	R-squared:	0.643			
Model:	OLS	Adj. R-squared:	0.643			
Method:	Least Squares	F-statistic:	1.299e+05			
Date:	Tue, 13 Aug 2019	Prob (F-statistic):	0.00			
Time:	22:34:12	Log-Likelihood:	-3.0277e+05			
No. Observations:	289046	AIC:	6.056e+05			
Df Residuals:	289041	BIC:	6.056e+05			
Df Model:	4					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]

Intercept	213.3827	272.719	0.782	0.434	-321.138	747.904
yearOfRegistration	-0.1014	0.135	-0.750	0.453	-0.367	0.164
powerPS	0.0087	2.18e-05	400.587	0.000	0.009	0.009
kilometer	-6.693e-06	3.65e-08	-183.439	0.000	-6.76e-06	-6.62e-06
Age	-0.1848	0.135	-1.366	0.172	-0.450	0.080

b. Model Comparison (Numeric Features):

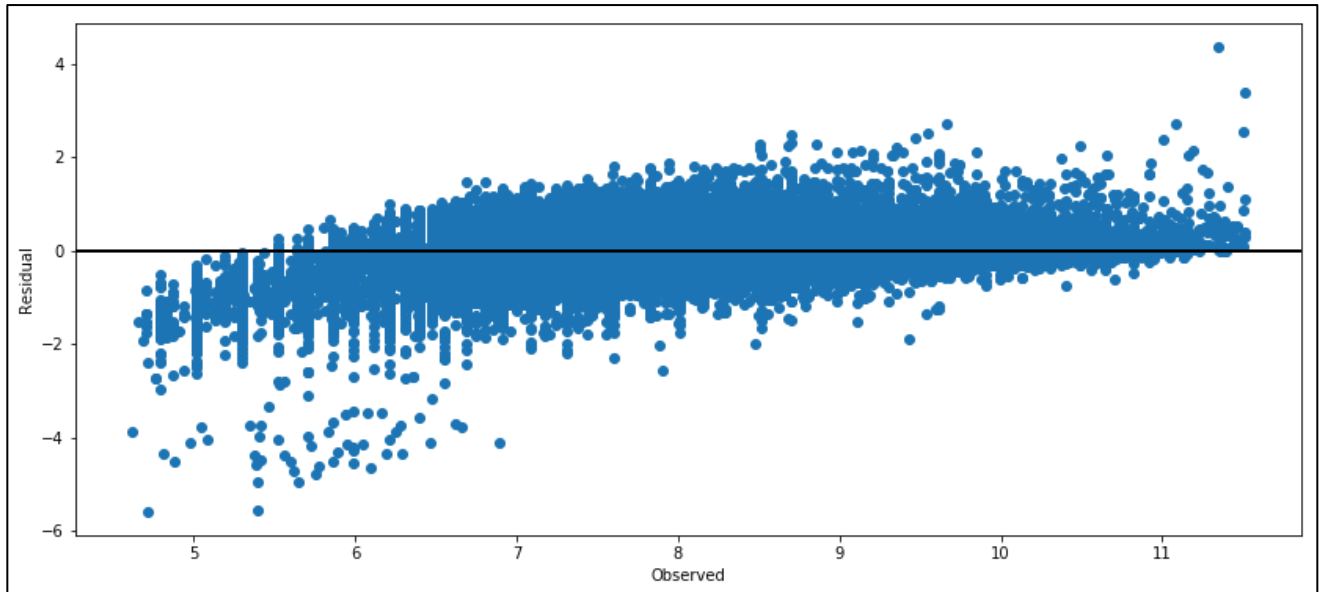
Model	R ² _Test	R ² _Train	Mean Squared Error (MSE) Test	Mean Squared Error (MSE) Train
Linear Regression	0.643	0.642	0.4755	0.4757
Ridge	0.640	0.640	0.4786	0.4786
Lasso	0.642	0.642	0.4759	0.4761
Random Forest	0.793	0.831	0.2740	0.2240

c. Model Comparison (With All Features):

For nonnumeric features I created dummy variables in order to use them in my models. The results the models are shown below.

Model	R ² _Test	R ² _Train	Mean Squared Error (MSE) Test	Mean Squared Error (MSE) Train
Linear Regression	0.776	0.779	0.2966	0.2935
Ridge	0.772	0.775	0.3026	0.2989
Lasso	0.642	0.642	0.4759	0.4761
Decision Tree	0.764	0.768	0.3138	0.3078
Random Forest	0.873	0.942	0.1683	0.0771

There is not a significant R^2 and MSE difference between our test and train data as seen in the tables. (There is a small difference in the Random Forest model, which means that our data needs more samples, which I would suggest as a future work). After applying nonnumeric features, the accuracy of our models increased. Best accuracy with the highest R^2 (0.873) and lowest MSE (0.1683) is the **Random Forest** model. After adding categorical data, the score increased from 0.79 to 0.88. And the residual plot after applying categorical data is shown below.



5. CONCLUSIONS:

The aim of this project is to predict the price of a used vehicle based on its features. In my model, after cleaning the data and applying the feature selections, I applied linear regression models with numeric features. The results were not very good with the highest accuracy of 64%. After that, with other categorical features, I created dummy features and applied them to the models. Random Forest showed the best performance after all the features applied to the model. I could manage to catch up **%88 accuracy** in predicting the price of a used car.

6. FUTURE WORK:

After adding categorical data to the model, the number of dummy variables increased. In order to get better results, I could do an effective feature selection, dimensionality reduction and hyper tuning to decrease the number of less effective features.

The data includes only the year of 2016. Thus, for creating more accurate models new data could be added including the years of 2017 and 2018 to create a more accurate model. On the other hand, a model that predicts 'how long a vehicle would stay active in the webpage before it is sold' could also be created with the accurate features.