Estimate your car price

Learn how you will be able to estimate your car price tomorrow

Graphical visualization of all car brands

“Our intelligence is what makes us human, and AI is an extension of that quality.”
– Yann LeCun Professor, New York University

A month ago, our AI teacher told us about the next project that we will have. Without any doubt, we were in awe… we were beginners in Artificial Intelligence, and we had to develop something smarter than a… self-drive car!

More seriously, after a dozen of courses about AI, my friend and I, had to take up a huge challenge: find a subject and develop an intelligent program about it. Of course, our teacher taught us the basis et we did some practical exercises, but after that, it was our turn to act.

First step: find a subject

I think that, like all projects, it’s one of the most difficult steps. Moreover, it’s hard to know if our idea will be reachable with our tiny level. But we like challenges, so we started to look in different websites, searching a dataset. Why a dataset? We were looking for a dataset because we needed, to start, lots of data to exploit. In a nutshell, the dataset is basically the subject we want to work on.

In a first time, we found a dataset about the actual epidemic Coronavirus. Unfortunately, it was useless. Data were not logic and even corrupted. I advise you, if you look for a dataset, consider its usability, really. But after a while, we finally found something relevant and useful : a dataset containing more than 250,000 cars, with lots of characteristics like the brand, the kilometres, the year of registration, the fuel, the price etc… an exhaustive dataset.

Here we are, now we were able to set up our idea. So basically, it was to create an intelligent program that would be capable to estimate a price of a car in function of the different parameters that we will enter. The program would be in Python and the final deliverable would be an Android app to use it properly. But stop, I don't want to give you too much hope… The app has not been created yet, we are working on the program to finalize it, and, when everything will be tested, we will develop the app. For now, let's talk about the Python Program in 4 steps:

1. The importation of the dataset
2. The Data Cleaning
3. The Data Visualization
4. The Model Training

The importation of the Dataset

First, let me use this part to tell you where we found the dataset and the Notebook we used.

We found the dataset in *Kaggle*. It's a website where you can find lots of datasets and challenging programs to develop, with prices to earn.

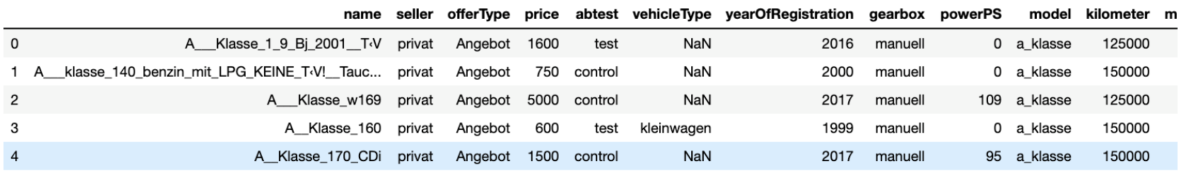
Here is the link of our dataset : https://www.kaggle.com/orgesleka/used-cars-database

After that, we decided to use Jupyter, a Notebook of Anaconda, to develop programs with Python. Then, the first thing we did in Jupyter was to import the dataset we just downloaded before. This is where it all begins!

The Data Cleaning

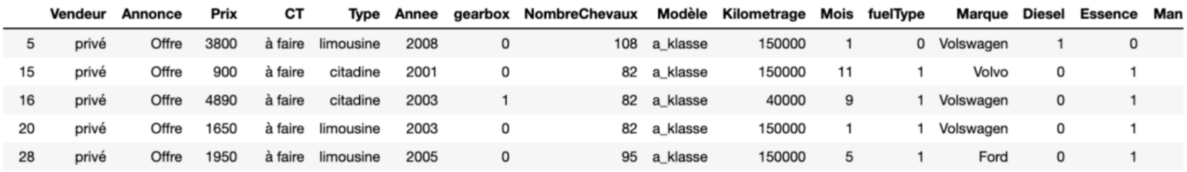
Now we have in our disposition the dataset, ready to be used. But you will see that when you find something on Kaggle, it will hardly be a dataset 100% useful… So this step has the aim to clean each data which is not necessary.

In our dataset, we first started by cleaning all empty cells, then we removed some details like cars which were registered before 2000, or the cars which have electrical power etc… We cleaned it as far as possible to obtain the best dataset for our purpose (and yes, we decided to work only on diesel and essence cars for now). Also, we traduced the names of some columns and cells because the dataset was written in German… Look how was the dataset before and after the Data Cleaning:



	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	m
0	A___Klasse_1_9_Bj_2001__T‹V	privat	Angebot	1600	test	NaN	2016	manuell	0	a_klasse	125000	
1	A___klasse_140_benzin_mit_LPG_KEINE_T‹V!__Tauc...	privat	Angebot	750	control	NaN	2000	manuell	0	a_klasse	150000	
2	A___Klasse_w169	privat	Angebot	5000	control	NaN	2017	manuell	109	a_klasse	125000	
3	A__Klasse_160	privat	Angebot	600	test	kleinwagen	1999	manuell	0	a_klasse	150000	
4	A__Klasse_170_CDi	privat	Angebot	1500	control	NaN	2017	manuell	95	a_klasse	150000	

Before the cleaning …



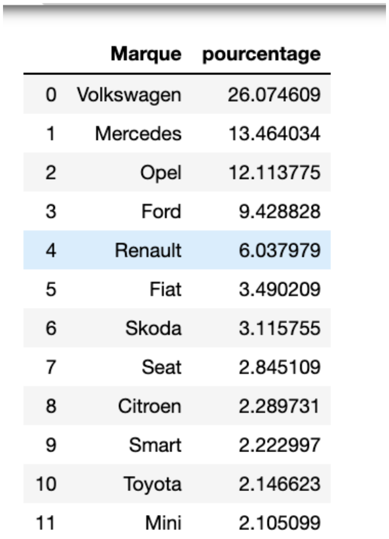
	Vendeur	Annonce	Prix	CT	Type	Annee	gearbox	NombreChevaux	Modèle	Kilometrage	Mois	fuelType	Marque	Diesel	Essence	Man
5	privé	Offre	3800	à faire	limousine	2008	0	108	a_klasse	150000	1	0	Volswagen	1	0	
15	privé	Offre	900	à faire	citadine	2001	0	82	a_klasse	150000	11	1	Volvo	0	1	
16	privé	Offre	4890	à faire	citadine	2003	1	82	a_klasse	40000	9	1	Volswagen	0	1	
20	privé	Offre	1650	à faire	limousine	2003	0	82	a_klasse	150000	1	1	Volswagen	0	1	
28	privé	Offre	1950	à faire	limousine	2005	0	95	a_klasse	150000	5	1	Ford	0	1	

After the cleaning ! (yes, I'm french in reality, but shuut :))

The Data Visualization

Now, we had to visualize our data by representing them on graphs or tables, just to give them a shape in our head and for the incoming program.

So, here, we started by creating a table just to see the percentage of each brand in our dataset, like this:



	Marque	pourcentage
0	Volkswagen	26.074609
1	Mercedes	13.464034
2	Opel	12.113775
3	Ford	9.428828
4	Renault	6.037979
5	Fiat	3.490209
6	Skoda	3.115755
7	Seat	2.845109
8	Citroen	2.289731
9	Smart	2.222997
10	Toyota	2.146623
11	Mini	2.105099

Percentage of each brand

Then, we created the graph which I put at the beginning of the article, with a big size of text for the main brands, and a smaller size text for the others, progressively.

We also represented in different types of graphs the data about the kilometres in function of the price, the brand in function of the price etc… I encourage you to check our GitHub where we put all the code!

This part of data visualization is just essential; it's cool to have a dataset cleaned, but if you don't visualize the data you're going to use… it's not so cool! Moreover, we use them in the next part: The Model Training.

The Model Training

The model training is the hardest part because it's the most technical of all. Here we must create our method to estimate the price, but which one?

Our topic uses the linear regression. In statistics, linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables. Basically, we want to represent the price with the parameters we saw before.

In linear regression, there are multiples methods from different libraries, like *Sklearn* or Ordinary Least Squares (OLS). These methods are very similar, but we decided to use OLS, it seemed to be more appropriated for a price estimation.

So we used a function using OLS (I let you see the Code on GitHub) in which we have to put some parameters of our car, and the function estimates the price!

Conclusion

We're done for now. This article tried to introduce with some details our project, especially how we implemented it. Like I said, the idea is still in development, the final purpose has not been reached yet. But when everything will be perfectly tested, an app will be released and will allow you to estimate your car price !