Documentation for Machine Learning Final Work

Luis Cabañero Gómez

Iván García Pulido

Problem explanation

The dataset to work with has information about used cars on sale on Ebay (of Germany). The purpose of this work is examining the data in order to find a relation between the field that indicates if a car has damage not repaired and somo other fields, like price or kilometers. Also, it is pretended to find what group of cars has more unrepaired damages.

Data description

The original database contains 371825 rows and 20 columns and its fields are the next:

- dateCrawled : When the data was obtained
- name: Car name
- seller: Private seller (sealer) or dealer.
- offerType: Offer (Angebot) or petition (Gesuch).
- price
- abtest: An internal Ebay field. Not very relevant
- vehicleType: Takes the next values: coupe (coupe), suv (Sports Utility Vehicle), kleinwagen (small car), limousine (sedan), cabrio (convertible), bus (minivan), kombi (van), andere (strange car).
- yearOfRegistration : Year of the car’s first matriculation.
- gearbox: Can take to values: manuell (manual) or automatik (automatic).
- powerPS: Car’s power in horsepower.
- model
- kilometer : number of kilometers the car has traveled. It has fixed values.
- monthOfRegistration : Month of the car’s first matriculation.
- fuelType: It can be: benzin (gasoline), diesel (diesel), hybrid, elektro (electric), lpg (gas), cng(gas), andere(another one).
- brand
- notRepairedDamage : The car has or not damages not repaired.
- dateCreated : Publication date of the advertisement.
- nrOfPictures : Number of pictures. This field is always 0 because an error fetching the data.

- postalCode: Seller’s postal code
- lastSeenOnline : When the ad was last seen by the crawler.

In order to work properly with the data, it has been cleaned and preprocessed. In this process it has been done the next modifications:

- Columns erased directly: nrOfPictures, abtest, dateCreated, postalCode, dataCrawled, name.
- Columns erased with a row selection before:
    - seller: The only ones taken are the private sellers.
    - offerType: The ones taken are the offer.
- Duplicate rows are dropped.
- Rows with empty fields are dropped
- Row selection:
    - price: greater equal than 200 and lesser 100000.
    - yearOfRegistration: dates greater equal than 1980 and lesser than 2017.
    - powerPS: power greater than 35 and lesser than 1000.
    - fuelType: erased andere.
    - model: erased andere.
- Combined year of registration with month of registration and lastSeen to create the field antiguedad which indicates the age of the car in months.

Algorithms applied

Before talking about the algorithms used, it is important to mention that the size data has been reduced randomly due to the algorithms couldn’t be executed in our machines. The algorithms used are the next:

- DBSCAN: In order to make an initial clustering dbscan has been applied to the fields: ‘price’,’kilometer’ and ‘antiguedad’. This has been done in this way to have clusters of similar cars. The clustering gave 56 clusters using the next parameters:
    - metric: “euclidean”
    - min_samples: 8
    - eps: 0.015

Once the clustering has been done, the next features has been defined for each cluster: number of cars, mean of powerPS, number of different models, number of different brands, mean of months, mean of price, mean of kilometer, proportions of broken cars, a feature of the proportion of cars using each fuel and a feature of the proportion of each type of car.

- Hierarchical: In order to get a second clustering over the previous one a hierarchical clustering has been applied. The hierarchical cluster wasn’t applied at first because there were too many data and this kind of clustering isn’t very efficient. To this clustering the only features used has been the power and the proportion of cars and fuel of each type. It has been used a euclidean metric and the complete-linkage method. The threshold was decided after seeing the dendrogram: 5.

- Decision tree: Finally, in order to get a way to get what cars have unrepaired damage according to its vehicle type, model, brand, gearbox and fuelType a decision tree was applied. If the tree isn't pruned, it takes to much to plot and it is too big to properly obtain conclusions.

Results and conclusions

The hierarchical clustering got 4 clusters. This table is a summary of it:

Cluster	Number of cars	Mean of the price	Mean of the kilometers	Mean of the months	Proportion of broken cars
1	122	37292.222	45500.0	28.577	0
2	100142	7636.979	44833.333	125.6485687	0.1012562
3	92	19159.506	83500	131.9713564	0
4	317	5566.843	22857.143	125.7897626	0.06624606

It can be noticed some relevant information:

- The first is that there is a cluster much bigger than the others, so it don't give so much information as it would be desired.
- According to the price and months it can be seen that the cluster 1 and 3 corresponds to high-end cars and they don't have any damage.
- The cluster 4 correspond to low-end cars and they have broken cars, but not too much.
- The most relevant information that can be extracted from this clustering is that using only the proportions of cars of each type of the previous clustering it can be selected some high-end cars.

The decision tree is capable to predict if a car is damaged according of the rest of the information about it, but only if it is large enough. If the tree is pruned to the point it can be interpreted by a human and it can be generated in a reasonable time it needs to be around 7 of max-depth and in this case, it doesn't predict very well the class.