Capstone Proposal

Rui Cao
Udacity

1 Domain Background

In the used car market, one important problem is to determine a used car's price. A seller always try to sell the car with a price as high as possible. Meanwhile, price of a car is often a very significant factor for the costumer's choice of buying. So, an effective estimation of price would provide very useful guide for both the sellers and buyers when dealing with a used car. Generally speaking, the determination of a used car's price depends on its brand, model, year of usage, and many other factors. This makes the determination of a used car very complicated. Therefore, a machine learning method would be proper to explore the hidden relation between those factors and the used car's price.

Once the model is established, it can provide some references to other problems as well. For example, evaluating the attrition of a current processed car, or predicting whether a car needs to be replaced.

2 Problem Statement

The problem is for a dataset crawled from ebay. The dataset provides every details of the used car ads posted on ebay within a period of time, from March 2016 to April 2016. Most of the details are presumably related the price determination. Moreover, most of those details can be quantified or treated in a quantitative way. Meanwhile, as the price appears in the format of a number, one way to compare the prediction and actual value is to take an absolute value of the difference between the log of these two quantities, i.e. the absolute log error. From all the above, it is expected that a prediction of the car's price can be made based on these information through the method of machine learning.

3 Datasets and Inputs

The dataset is extracted from a used car database.

The dataset consists 371528 rows of data. Each row represents the data of a used car ads and contains following fields:

dateCrawled The date when the ad was first crawled. All other fields comes from the ad on this date.

name Name of the car, it may contain information of brand, model, etc.

seller Private seller or dealer.

offerType Offer type is the same for all data, so this field is useless.

price The price on the ad to sell the car. This is the quantity needs to be estimated. So this field will removed from the data.

abtest A ebay-intern variable.

vehicleType One of the eight vehicle categories.

yearOfRegistration The year that the car was first registered.

gearbox Type of the car's gearbox, manual or automatic.

powerPS Power of the car in PS.

model The car's model.

kilometer Number of kilometers the car has driven.

monthOfRegistration The month that the car was first registered.

fuelType One of the seven fuel categories of a car.

brand Brand of a car.

notRepairedDamage If the has a damage which is not repaired yet.

dateCreated The data for which the ad at ebay was created.

nrOfPictures Since there is a bug in the crawler, all numbers in this field is 0. So the field is useless.

postalCode The place in Germany where the car is located in.

lastSeenOnline The time that the crawler saw this ad last online.

4 Solution

First of all, fields of 'offerType', and 'nrOfPictures' are useless because they are having the same value for all rows. The field 'price' will be removed. Then the date fields will be transferred into numerical forms by calculating the days from a fixed date, for example, March 1st, 2016. Most text and categorical fields will be transferred into numerical forms using one-hot encoding. The 'name' field will be preprocessed, to extracted only brand or model of the car, then it could be used to repair the missing or incorrect value in the field of 'brand' and 'model', it may also be used to repair missing value in 'vehicleType' by comparing with other rows. However, the information extracted from the field 'name' will not be used as input of the model. After these, all needed information are in numerical form. Then a supervised machine learning model will be trained and is expected to make reasonable price predictions. A multi-layer neural network will be the first candidate because it's easy to this model to catch the nonlinear relations between the features and the outcome, and thus the model can make more precise predictions. A random forest model or mini batch gradient decent model may also be considered if the multi-layer neural network model doesn't work.

5 Benchmark

A simple mean value model will be used as a benchmark. The model takes the prices of all records from the dataset and evaluates their mean. The model will use this mean value as the prediction for any car's price.

6 Evaluation Metrics

Since the price of a car appears to be a number. The actual price advertised can be directly compared with the prediction price. The model is evaluated using **mean absolute log error** between the two prices, which is:

$$E = \frac{\sum_{i=1}^{N} |\log(P_p) - \log(P_a)|}{N}$$

where P_p is the prediction of the car's price, P_a is the car price advertised.

7 Project Design

First the useless fields will be removed. The 'price' field will be extracted as target variable. Then I'll deal with abnormal values in the fields. All possible abnormal values will be repaired if its value is indicated in other fields or records, or will be taken to be None if it makes sense. If both of the ways don't work, that ad record will be discarded. For example, the 'NaN' vehicleType records will be repaired by comparing to other records with the same car brand and model and a valid vehicle type. If no 'vehicleType' is available, then value 'NaN' will be kept. While the record with 0 price will be discarded because it is meaningless to be an ad. After the data is cleaned. Some visualization and statistics of the data will be present to future check if the data is in a skewed distribution or some fields need to be normalized. Then the text values will be quantified using one-hot encoding. Once all needed information are in numerical forms, they can be analyzed using some mathematical tools, for example, a PCA will be taken to see if there are highly linear relations between some features. After all the above, I can decide which features to be taken as input. Finally, I'll train a multilayer neural network model and tune its hyper parameters to improve its performance. If its performance is desirable, I may consider other type of models mentioned in the solution section.