

Price Prediction: An Used Car Case Study

Haofan Wu
Department of CSE, UCSD
PID: A53237402
haw013@ucsd.edu

Changliang Cao
Department of CSE, UCSD
PID: A53235075
chc506@ucsd.edu

Shuang Li
Department of ECE, UCSD
PID: A53242288
shl668@ucsd.edu

Pengfei Song
Department of CSE, UCSD
PID: A53246932
pesong@ucsd.edu

Abstract

In this project, we consider the problem of predicting price of the used car from eBay used car database. The project involves finding out the appropriate features for the task. After that, we train these features using machine learning techniques such as Linear regression, Bayesian regression, K-NN regression, Decision Tree regression, Gradient Boosting regression and Random Forest regression. For the price prediction task, Gradient Boosting regression and Random Forest regression performed the best, yielding a R-square of 0.8247 and 0.8384 respectively.

Keywords

Linear regression, Bayesian regression, Decision Tree regression, k-NN regression, Random Forest regression, Gradient Boosting regression, MAE, MSE, R square

1 Introduction

In recent years, with the frequent update of autos and some other reasons, more and more cars are experiencing an early retirement. To make full use of these wasted resources, a brand new commercial chain has appeared to collect those used cars in relatively good condition and sell them online. But the problem is for a new user who registered to sell his/her used car, but might not know the appropriate price to label it according to its qualities and condition.

The price of a used car mainly depends on many features such as type of vehicles, time of registration, brand and model etc. Therefore, our goal is to determine the relevant features by performing exploratory analysis

on the dataset. After extracting all the relevant features, we are planning to use six different machine learning techniques to train these features and to locate the price level of the item. They are as follows:

- 1) Linear regression
- 2) Bayesian Regression
- 3) k-NN regression
- 4) Decision Tree Regression
- 5) Gradient Boosting regression
- 6) Random Forest regression

The analysis and comparison of different models and reasoning about them is displayed in the subsequent subsections.

2 Dataset and Tasks

2.1 Basic Statistics and Properties

The dataset we used for our task is a part of eBay used car database and we download this part of data from kaggle. This set includes information about 370,000 used cars and their business information on the eBay website. These fields are as follows:

- 1) dateCrawled : when this ad was first crawled, all field-values are taken from this date
- 2) name : "name" of the car
- 3) seller : private or dealer
- 4) offerType
- 5) price : the price on the ad to sell the car
- 6) abtest
- 7) vehicleType
- 8) yearOfRegistration : at which year the car was first registered
- 9) gearbox

- 10) powerPS : power of the car in PS
- 11) model
- 12) kilometer : how many kilometers the car has driven
- 13) monthOfRegistration : at which month the car was first registered
- 14) fuelType
- 15) brand
- 16) notRepairedDamage : if the car has a damage which is not repaired yet
- 17) dateCreated : the date for which the ad at ebay was created
- 18) nrOfPictures : number of pictures in the ad (unfortunately this field contains everywhere a 0 and is thus useless (bug in crawler!))
- 19) postalCode
- 20) lastSeenOnline : when the crawler saw this ad last online

2.2 Data Distribution Analysis

Since there are so many features in this dataset, we need to drop some useless features. We have two types of data need to drop, one is the data can not be used in prediction, such as nrOfPicture (since the value of each element is 0). The other is the data that is not related to price, for example, empirically, dateCrawled, name, lastSeen and postalCode are nothing to do with price. After this analysis, we dropped the following fields: seller, offerType, abtest, dateCrawled, nrOfPictures, lastSeen, postalCode, dateCreated, name.

The remaining features are as followings: vehicleType, yearOfRegistration, gearbox, powerPS, model, kilometer, monthOfRegistration, fuelType, brand and notRepairedDamage. We analyze these features' distribution first and use Histogram and Word Cloud to describe the distribution:

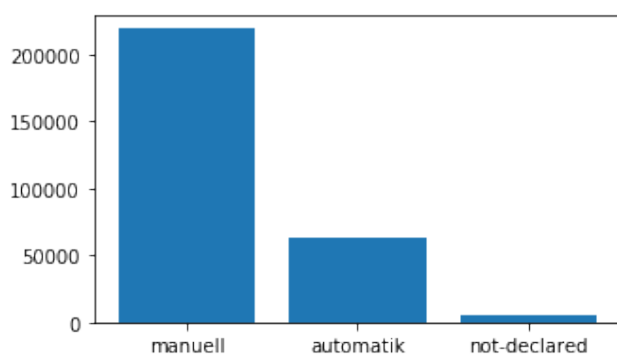


Fig.1 gearbox distribution

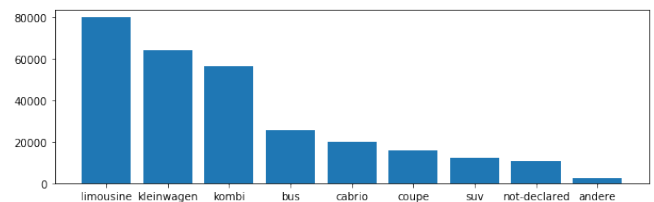


Fig.2 vehicleType distribution

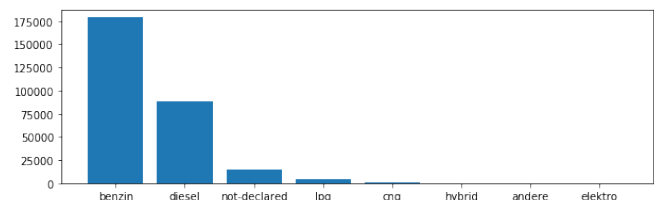


Fig.3 fuelType distribution

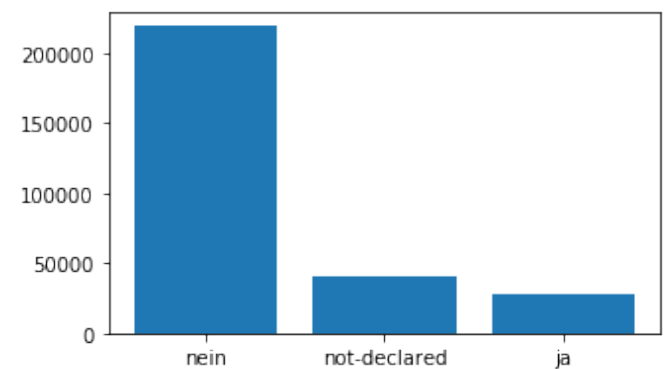


Fig.4 notRepairedDamage distribution

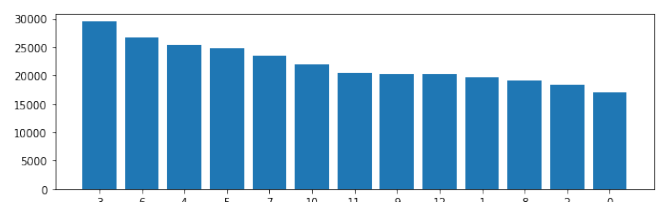
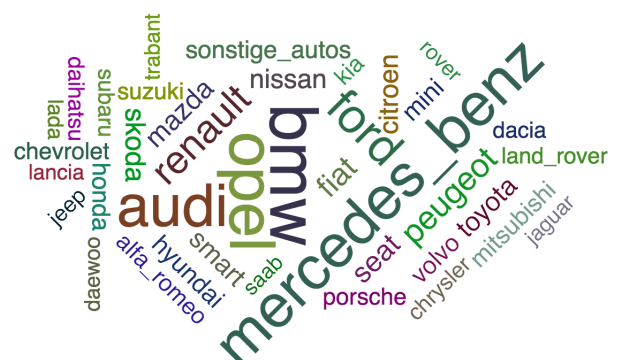


Fig.5 monthOfRegistration distribution



[illegible]

Line Graph


Year

Year	Deaths
1950	0
1954	0
1958	0
1962	0
1966	0
1970	0
1974	0
1978	0
1982	0
1986	0
1990	0
1994	0
1998	14,000
2002	16,000
2006	17,000
2010	10,000
2014	4,243

yzing these features’ distribution, we find do not fit the empirical facts, for example, some items is 0. So we selected data the fact, which are yearOfRegistration 2016 (since this dataset is crawled in 2016), to 150,000 and powerPS from 10 to 500.

So far, we can identify a predictive task which is to predict the price of used cars.

2.4 Features Versus Prices



A scatter plot showing the number of nodes in the network over time. The x-axis represents time from 0 to 500, and the y-axis represents the number of nodes from 0 to 150,000. The plot shows a dense cloud of blue dots, indicating a high frequency of node additions and deletions. The number of nodes generally increases over time, with a significant peak around time 200, reaching approximately 140,000 nodes. After this peak, the number of nodes fluctuates between 50,000 and 100,000.

same as the feature of yearOfRegistration, the month of registration also has an effect on the selling price:

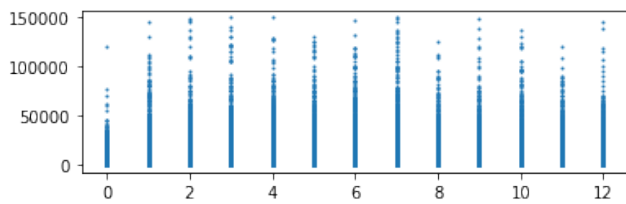


Fig.12 Prices versus monthOfRegistration

From figure, we can find that the car registered in summer has higher price than other months, so monthOfRegistration needs to be considered.

2.4.5 brand

In modern life, brand plays a very important role in price setting, so we analyze the relation between brands and prices, with the index of brands as x axis.

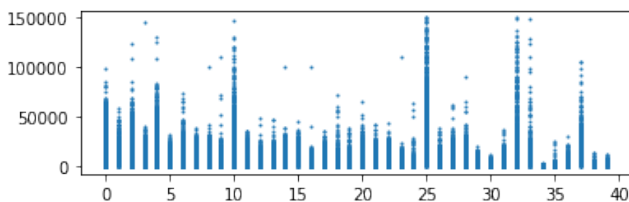


Fig.13 Prices versus brand

In the result, we find that index 25, 32 and 10 have the highest price, and these three brands are porsche, sonstige_autos and mercedes_benz.

2.4.6 model

Same as brand, model will affect the price, and the relation is shown as follows:

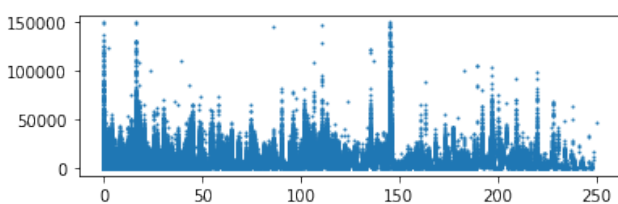


Fig.14 Prices versus model

Models with high price such as index 15 and 140 are arosa and qashqai.

2.5 Data cleaning and pre-processing

After analyzing the relationship between features and price, we decide to choose the following features: vehicleType, yearOfRegistration, gearbox, powerPS, model, kilometer, monthOfRegistration, fuelType, brand and notRepairedDamage.

To create the feature matrix, we need to handle features that is not numeric. We use one hot encoding to solve this problem.

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction. The key thought is to make data one-hot. (one-hot is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0)).

Take vehicleType as an example, there are 9 kinds of vehicles and we denote each as 000000001 format. Using this denoting method we assume that each feature that is not numeric is independent, but we find that brand and model have something duplicated, so we combine these two columns (eg. mercedes_benz_e_klasse) and encode them after the combination.

After the encoding, we got a feature matrix that contains only numbers and then we can use models to predict the price.

3 Models

3.1 Linear Regression

The linear regression model was used as a baseline. We encountered over-fitting with this model for the features actually did not take on the linear trend and the linear model was just a rough estimation of the simulation of the relations between those features.

3.2 Bayesian Regression

Bayesian linear regression method assumes that the error of the regression model is independent and identically normally distributed. In the Bayesian approach, the data are supplemented with additional information in the form of a prior probability distribution. The prior belief about the parameters is combined with the data's likelihood function according to Bayes theorem to yield the posterior belief about the parameters of this model.

3.3 K-nearest neighbors Regression

K-nearest neighbors method is a non-parametric

technique and it stores all available cases then predicts the numerical target based on a similarity measure. This model can be optimized by tuning the parameter K. In general, a large K value is more precise as it reduces the overall noise; however, the compromise is that the distinct boundaries within the feature space are blurred.

3.4 Decision Tree Regression

This model uses a decision tree (as a predictive model) to go from observations about an item (branches) to conclusions about the item's target value (leaves). The tree can be learned by splitting the source set into subsets based on an attribute value test. This model was selected for its edge of handling large datasets, and its feature selection where additional irrelevant feature will be less used so that they can be removed on subsequent runs. But the decision tree model is less accurate than other models and a small change in the training data can result in a big change in the tree, and thus a big error in final predictions.

3.5 Gradient Boosting Regression

Starting with a very weak model, usually the mean of output, the algorithm reconstructs a new model by adding a estimator to provide a better model. This model selects the loss function instead of least-squares to minimize the difference between predictive values and actual output values based on the observation that the residuals for a given model are negative gradients of the square error of the loss function. Concisely, this model turns weak predictors to better ones by learning from the mistakes in previous steps and better in next steps by boosting the importance of the incorrectly predicted data points.

3.6 Random Forest Regression

Random forest is an ensemble learning method for regression. This method constructs a multitude of decision trees at training time and outputs the mean prediction of the individual trees. Comparing to single decision tree, this method combines "bagging" idea and random selection of features to avoid over-fitting on the training set. In addition to constructing each tree using a different bootstrap sample of the data, random forests change how the regression trees are constructed. Each node is split using the best among a subset of predictors randomly chosen at that node.

4 Results and Conclusion

In order to compare the performance of the six regression models, we draw histograms of their MAE, MSE and R-Square. According to these graphs, we come to the following conclusions:

1. Random Forest model performs best of the six in general while linear regression and Bayesian regression lead to worst results. Besides, the other three models obtain relatively satisfactory results.
2. Ensemble learning techniques like Random Forest and Gradient Boosting generally performs better in datasets similar to the used car dataset according to our experiments and literature review.
3. Our results can provide a favorable reference about the appropriate price for sellers of second-hand cars and for customers who are planning to purchase a used car.
4. By comparison of two different non-numeric feature representations: simply encoding different features with sequential integers and one hot encoder, we found that the latter one led to more accurate results.
5. Interpretation of the parameters in models:
 - 1) Knn (Parameter k): a large k value is more precise as it reduces its overall noise while it will blur the decision boundaries within the feature space.
 - 2) Decision Tree (Parameter depth): deeper decision tree will make use of more attributes in the feature vectors, thus improve the performance on the train dataset. Meanwhile, it could damage the performance on the test data due to overfitting.
 - 3) Random Forest: there are two parameters in the random forest method: the number of variables in the random subset at each node and the number of trees in the forest. The former is similar to the Parameter depth in the Decision Tree, while the larger value of the latter parameter will further eliminate the noise of each individual tree.
6. The features we use in the prediction contains many categorical fields, thus methods like Decision

Tree and Random Forest are more suitable. Besides, Ensemble learning techniques like Random Forest combine multiple decision tree's performance to avoid individual regressor's bias.

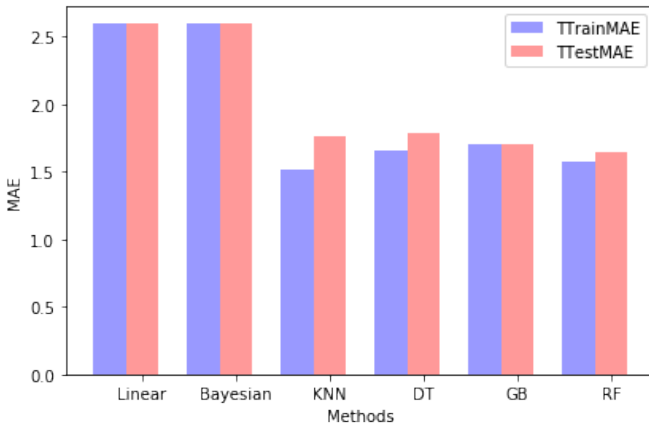


Fig.15 MAE of Six Models

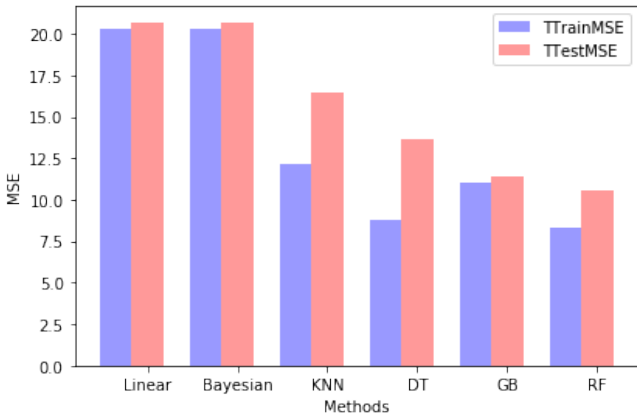


Fig.16 MSE of Six Models

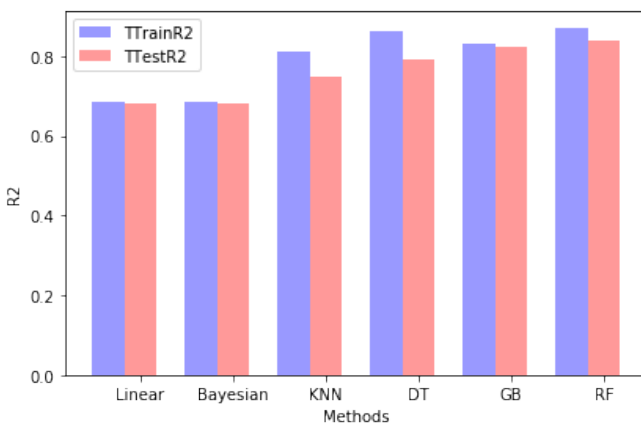


Fig.17 R2 of Six Models

5 Literature

5.1 About Dataset

The dataset we used for our task is a part of eBay used car database and we download this part of data from

kaggle. This set includes information about 370,000 used cars and their business information on the eBay website. We preprocessed the dataset by dropping some useless fields and removing the duplicate and out-of-reasonable-range data. Then we divided our dataset into training set and test test, analyze the useful features with different models.

5.2 Similar Dataset

One example of the similar cars datasets is scraped from Edmunds and Twitter. The dataset has more than 11,915 car samples and its fields include make, model, year, engine fuel type, engine HP, engine cylinders, transmission type, driven wheels, number of doors, market category, vehicle size, vehicle style, highway MPG, city mpg, popularity, MSRP. Most studies working on the cars dataset use Linear regression, Ridge regression and Lasso regression model to predict the price of new and old cars, which usually have relatively lower performance. Some previous studies use random forest regression and gradient boosting regression model to predict the price of the car, which have the better performance. As mentioned before, we include Linear regression, random forest regression and gradient boosting regression in our choices of models and get the similar results that random forest regression and gradient boosting regression perform better than linear regression. These studies also find that the most important features in determining the price of cars are the engines horsepower, engines fuel type, engine cylinder, model, make and year, which is similar to the features we choose, like fuel type, model, year and so on.

5.3 State-of-the-art Method

From other studies on the predictions of food, house and hotel price, we find a state-of-art model called eXtreme Gradient Boosting, short for XGBoost. It has faster execution speed, better model performance and can be used in task for price prediction. The implementation of XGBoost is engineered for efficiency of compute time and memory resources. The design goal is to make the best use of available resources to train the model. The key algorithms XGBoost implementing include Sparse Aware, Block Structure and Continued Training. We attempt to run XGBoost model with default parameters on the features we extracted to predict the used cars price and do get better performance than common linear

regression. The R^2 score on the training set is 0.8286 and the score on the testing set is 0.8241. After simply modifying the parameters of XGBoost, we get higher score, 0.9510 on the training set and 0.8827 on the testing set.

5.4 Compared with our model

In conclusion, we find several similar car datasets and take one of them as instance. Previous studies on this dataset find that the most important features in determining the price of cars are the engines horsepower, engines fuel type, engine cylinder, model, make and year, which is similar to the features we choose, like fuel type, model, year and so on. And previous studies working on car price prediction usually use Linear regression, Ridge regression, Lasso regression, random forest regression and gradient boosting regression model. Study shows that random forest regression and gradient boosting regression have better performance than linear regression, which is similar to our results. Besides, after analyzing the studies on the price prediction of other things like food, house and hotel room, we find a state-of-art model called XGBoost, which proves to have faster speed and better performance on our training set and test set. And its result is better than the popular models used for price prediction.

6 Reference

- [1]<https://www.kaggle.com/orgesleka/used-cars-database>
- [2]https://en.wikipedia.org/wiki/Linear_regression
- [3]https://en.wikipedia.org/wiki/Bayesian_linear_regression
- [4]https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- [5]https://en.wikipedia.org/wiki/Decision_tree_learning
- [6] https://en.wikipedia.org/wiki/Gradient_boosting
- [7]https://en.wikipedia.org/wiki/Random_forest
- [8]<http://www.bios.unc.edu/~dzeng/BIOS740/randomforest.pdf>
- [9]<https://www.kaggle.com/CooperUnion/cardataset>
- [10]<https://en.wikipedia.org/wiki/Xgboost>
- [11]<https://www.kaggle.com/mburakergenc/predictions-with-xgboost-and-linear-regression>