Used Car Price Prediction

Yu Zhang
University of California San Diego
A98069731
yuz083@eng.ucsd.edu

Jing Liang
University of California San Diego
A53213765
jil884@eng.ucsd.edu

Shuang Wan
University of California San Diego
A53211559
s7wan@eng.ucsd.edu

ABSTRACT

In this assignment, we compare methods of K-means clustering, linear regression, and random forest in predicting the price of used car based on mean squared error (MSE) and variance score. We select the most effective features in the data according to cross-correlations between each feature, which gives a much better performance than the baseline solution. As a result, with the same features selected in the model, we find that random forest weighs better than K-means and linear regression in this price prediction task.

Keywords

used car price prediction; K-means clustering; linear regression model; random forest model

1. INTRODUCTION

Automobile business is always one of the most popular industries nowadays. People have high demands in changing their cars in a regular basis, especially in United States. Thus, there is a rapid growth in the pre-owned car value lookup websites, like Kelley Blue Book. People can estimate price of their cars if they want to sell them, or they can look up the price if they want to buy pre-owned cars. Most of these website ask the brand, year, model, and car type to predict the price, whereas we think there are more features that would affect the price. Based on the dataset we use in this assignment, we finally find that there are seven features that would make considerable effects on the price.

2. DATASET

The data we use is from the pubic datasets on Kaggles [1]. It is a cvs file and originally has 371,528 entries. We use 'price'as prediction and choose effective features from the rest of contents in the data.

2.1 Basic Properties

The followings are the features that are given in the dataset:

dateCrawled : when this car was first crawled, all field-values are taken from this date.

name: "name"of the car

seller: private or dealer

offerType: Angebot or Gesuch

price: the price to sell the car, in USD

abtest: control or test

vehicleType: limousine, kleinwagen, or kombi etc.

yearOfRegistration: at which year the car was first registered

gearbox: manuell or automatik

powerPS: power of the car in PS

model: specific model of each brand

kilometer : how many kilometers the car has driven

monthOfRegistration: at which month the car was first registered

fuelType: diesel or benzin

brand: volkswagen, audi, or chrysler etc.

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: location of the car

lastSeenOnline: when the crawler saw this ad last online

2.2 Exploratory Analysis

We extract some useful features that related to the price prediction. Their detailed information are as following:

price range: 0 - 181000 (in USD)

range of PowerPS: 0 - 12512

year range: 1000 - 9999

kilometer range: 5000km to 150000km

number of fuel type: 7

number of brand type: 40

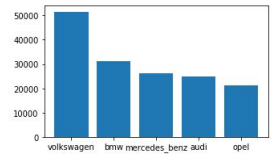
number of gearbox type: 2

From the information above, it is obvious that some ranges are not reasonable. Therefore, we need to drop some extreme and error data samples in order to obtain a better performance. These outliers are dropped in data preprocessing. The following histograms are the distribution of the significant features after the data preprocessing.

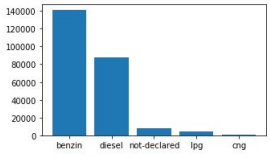
From these histograms below, we observe that most cars are Germany brands with manuell gearbox. In addition, all cars are mainly in two types, benzin and diesel. Assume that the owners registered their cars right after the purchases, most cars are more than ten years old. Power PS is around the medium range; only a few people are looking for the speedy experience with the used cars. All these features give an expected distribution; thus, we consider all these features as important features and will use these in our model.

From Figure 1 we can see the cars with name length smaller than 10 letters, larger than 40 letters and smaller than 60 letters have higher price than cars with other name lengths. The explanation is for some brands like BMW, it use number series to name their car, so the total name length is quite small. Mostly, the longer the name is, the more de-

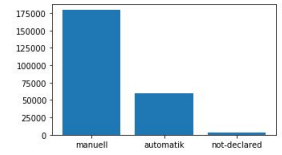
tails are contained, so the price is higher. If the name of the car is too long, it is still not good for the price.



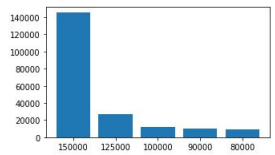
Distribution of top brands



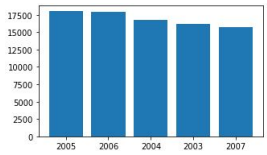
Distribution of fuel type



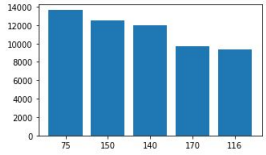
Distribution of gearbox



Distribution of kilometer



Distribution of year of reg.



Distribution of Power PS

3. PREDICTIVE TASK

3.1 Data Preprocessing

We follows these 7 steps to deal with our raw dataset:

1. shuffle the data.
2. delete irrelevant or less influential features with respect to the task of price prediction: 'dateCrawled', 'seller', 'offerType', 'abtest', 'notRepairedDamage', 'nrOfPictures', 'lastSeen', 'postalCode', 'deteCreated', 'model', and 'monthOfRegistration'.
3. drop outliers and keep samples which satisfy the following conditions: (1) the year of registration is from 1980 to 2016(neither too new nor old. (2) price is from 1000 to 150000. (3) driving distance is from 5000 to 200000. (4)powerPS is from 50 to 350.
4. delete the duplicates of sample.
5. delete all the sample which contains NAN.
6. numerate the string features to float: 'fueltype', 'brand', 'gearbox'
7. tune the right-skewed sale price by using logarithm. Figure 2 shows how distribution of price changes with this process. It shows that after the filtered data gives a Gaussian-like distribution.

After doing this, there is 61 percent of entire data left, which is around 22,000 samples. For these 22k data samples with complete information, we set 80 percent to training set and the rest 20 percent to test set. And for these 80 percent training set, we split another 20 percent training samples to build a validation set. As a result, we get training : validation : test = 6.4 : 1.6 : 2.

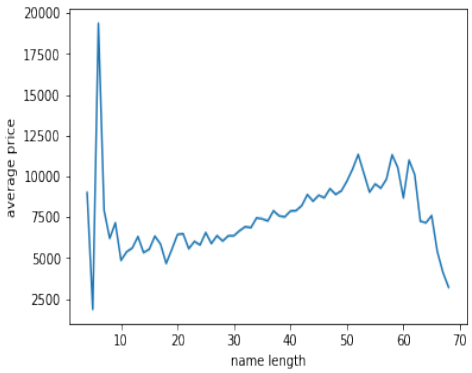


Figure 1: average price vs name length graph

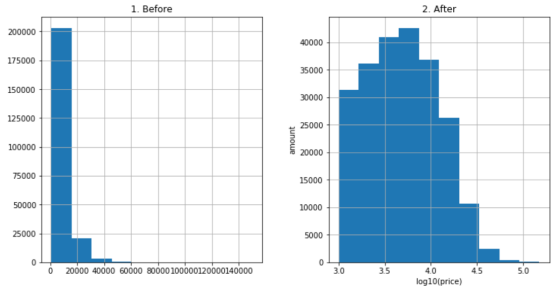


Figure 2: Histogram of price before and after processing

3.2 Feature Selection

In order to find the most significant features, we calculate the cross-correlation between each feature and price. We select features with high cross-correlation shown in Figure 4, and 0.1 is our threshold. Finally, the features we are going to use in both models are: 'yearOfRegiestration', 'powerPS', 'namelen', 'fuelType', 'kilometer', 'gearbox', 'gearbox', 'gearbox', and 'brand'. Features we selected matches to our exploratory analysis in the previous selection.

4. MODEL DESIGN AND SELECTION

4.1 Baseline

The baseline of used car price prediction is only using 3 features which are car brand, the year of registration and kilometers the car has driven.

Table 1: results of baseline

	training data	validation data	test data
MSE	0.08630	0.08635	0.08670
Var. score	0.42507	0.43011	0.42011

4.2 K-means Clustering

K-means clustering is a classification method which matches a data point to a cluster number. After matching, We can describe the data point by its cluster membership. At first, the number of clusters needs to be initialized and cluster centroids are chosen randomly. The process of K-means clustering is iterative which means date points are assigned

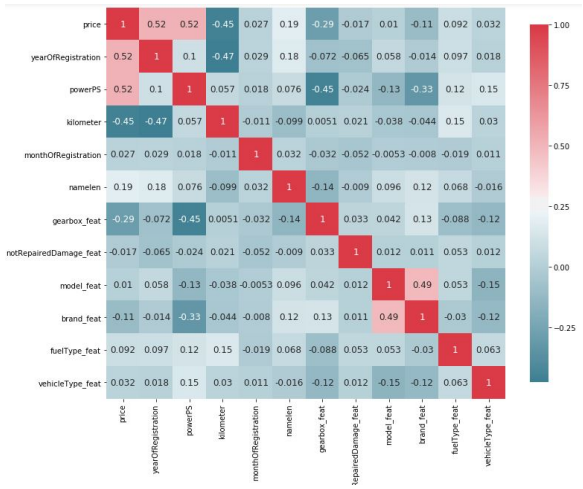


Figure 3: Cross-correlation

to theirs nearest centroids in each iteration and centroids change accordingly. It will converge to the state that minimizes the reconstruction error.

In this prediction task, the used cars can be grouped into k clusters based on their features. Then, we compute the average price for each cluster and regard it as the price that we predict. The assumption that we made here is that, cars with similar features also have similar values.

After training with training set, k cluster centroids and their average prices are obtained which are all we need to predict the price. And the best value for k is chosen with the help of validation set.

4.3 Linear Regression

Regression is one of the simplest approaches to learn relationships between features and labels. Linear regression assumes a predictor of the form $X\theta = y$, where X ,θ and y denotes matrix of features, model parameters and labels respectively. Our goal is to solve for θ which can minimize the mean square error between real labels y and predictions that is obtained by $X\theta$,i.g.

$$argmin_\theta = \frac{1}{N} \|y - X\theta\|_2^2 + \lambda \|\theta\|_2^2$$

The last term with λ plays the role of penalizing model complexity during training. If λ is too small, the risk of overfitting to the training set is high. And a too large λ will lead to naive model.

Gradient descent is the way to obtain the best θ. θ is first initialized randomly and updated iteratively by this equation $\theta = \theta - \alpha f'(\theta)$ until convergence.λ is not directly optimized in the process of gradient descent. Its best value is chosen according to the performance of validation set on different λ values.

4.4 Random Forests

Random forests builds many decision trees in one model. Each tree is a classification or a regression model, and outcome is the tree with the most votes. Random forests control the over-fitting problem caused by the a large number of decision trees; hence, the accuracy of the prediction is maximized.

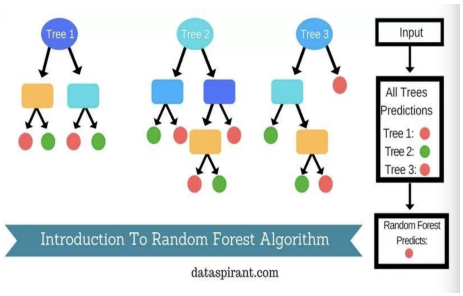


Figure 4: Random Forests Algorithm

We use the decision tree learning to perform random forests[2]. Each tree in the ensemble tree sets is created by the sample replacement, like AdaBoost. When build the branches between each ensemble tree, it does not choose the best path between two samples but creates the best path among a random selection. The tradeoff for the randomness is the increase in bias of a single tree. However, since the decision was made by taking the averaging between the sub-samples, so the variance scores decreases significantly and hence is a better model choice.

4.5 Final Model Selection with Optimization

Based on MSE and variance scores, we choose random forests as our final model, since it gives a much better performance than K-means and linear regression.

For random forests model, we implement different number of trees to see the prediction performance. And we find that MSE drops significantly at a certain tree numbers, and then MSE fluctuates around a certain value due to the randomness. Since our training data set is large, around 14,000 samples, we need to consider an appropriate large tree numbers to obtain the best performance when implementing the data; otherwise, some features would be missing.

5. RESULT

We use MSE and variance score to measure different methods. Suppose we use $\vec{x}$ to represent feature value, y to represent actual price, $\hat{f}(\vec{x})$ to represent price prediction. N is the number of samples in dataset.

$$score = 1 - \frac{u}{v}$$

$$u = \sum (y - \hat{f}(\vec{x}))^2$$

$$v = \sum (y - \frac{1}{N} \sum y)^2$$

The more complex the model $\hat{f}$ is, the more samples will be captured, hence the variance will be larger.

5.1 K-means Clustering

Figure 5 shows how MSE changes with the number of clusters. It's obvious that, MSE decreases quickly when k increases from 10 to 200. After k becomes bigger than 450, the curve tends to be flat. Since computation time is proportional to the value of k, there's no need to choose a bigger k to get a better accuracy. With k = 450, MSE on the test set is 0.05774 which is much better than that of the baseline.

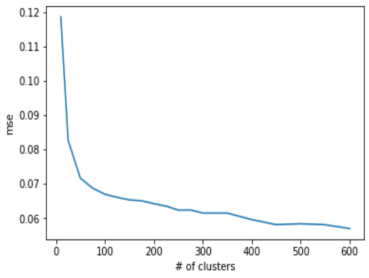


Figure 5: MSE vs number of clusters (k)

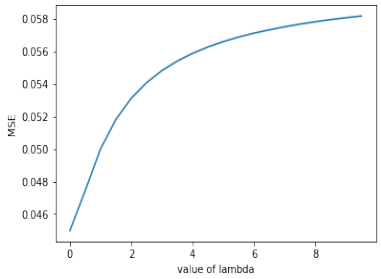


Figure 6: MSE vs value of λ in training set

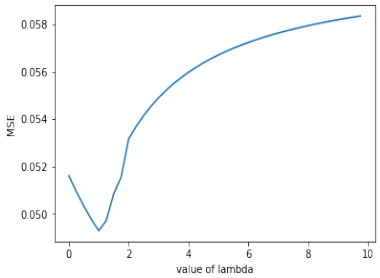


Figure 7: MSE vs value of λ in validation set

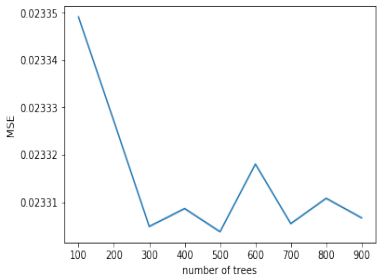


Figure 8: MSE vs number of trees in training set

5.2 Linear Regression Model

Figure 6 shows the MSE vs value of λ in training set.As λ goes up, which means we have more penalty for model being complex, the MSE value also increases. From Figure 7, we can see the sweet spot is at $\lambda = 1$. With $\lambda = 1$, the MSE of test set is 0.05039, the variance score is 0.66297. The price is preprocessed by log10.

Table 2: features and corresponding values

constant	-2.34123455e+01
name length	3.83698131e-05
year of registration	1.36451748e-02
gearType	1.49903898e-02
powerPS	3.47831076e-03
kilometer	-4.04099845e-06
brand	-3.11928351e-03
fuelType	-1.26400715e-01

From Table 2 we can conclude name length, year of registration, gearType and powerPS are positive correlated to price while kilometer brand, fuelType are negative correlated to price. Intuitively, the longer the name is, the more details it contains, the higher the price is. The newer and larger powerPS is, the more valuable the car is. The longer distance, the lower price.

5.3 Random Forests

We change the number of estimators to tune our model. Number of estimators is the number of trees we want to build before taking the maximum voting or averages of predictions. Figure 8 shows the MSE vs number of estimators we used for training set. Higher number of trees will give us better performance, so the envelope in Figure 8 decreases as number of trees goes up.

Figure 9 shows the MSE vs number of estimators we used for validation set. In general, we can see the MSE of validation set is larger than that of training set. Since higher number of estimators will make codes run much slower, considering our processor, we choose 700 as number of trees in our model finally.

5.4 Result Comparison

Table 3: results on test set

	baseline	K-means	Regress	Rand. Forests
MSE	0.08670	0.05774	0.05039	0.02454
Var. score	0.42011	0.61378	0.66297	0.83587

Table 3 shows the overall result on test set with different prediction methods that we implemented. MSE measures the accuracy of our prediction and it's explicit that it decreases as we use a better method which matches our expectation. As for the variance score, the best possible score is 1.0 and it can be negative when the model is arbitrarily worse. A constant model that always predicts the expected value of the price, disregarding the input features, would get a score of 0.0. According to the variance scores in the table, random forests has the highest value which implies that this model best fits data in the test set.

6. RELATED LITERATURE

The data we use is from the public dataset from Kaggle. It is originally collected with Scrapy from Ebay-Kleinanzeigen in German. Someone translated this dataset, and basically we have no idea that the price lists in the data is in USD or EURO; however, we do not think that currency unit would cause a big problem here, since in another perspective of standing, we are predicting how much amount of currency

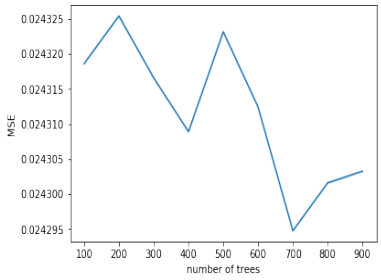


Figure 9: MSE vs number of trees in validation set

the car would worth in the dataset; the currency unit is in consistency.

We think we have enough features to predict the used car price in this dataset, and definitely other pre-owned car price checkout websites have their own dataset to estimate the used car price for clients. In addition, there are tons of similar price prediction datasets such as housing price prediction and diamond price prediction etc. For example, for housing price prediction, location, area, number of bedrooms and bathrooms, and level of decoration are all possible features, and as for diamond price prediction, cut, color and clarity might be the significant features.

We consider the price prediction is a kind of rating prediction, since it is in numerical value. But here, we do not have any customer-car pairs, so recommendation systems is not applicable in this dataset; thus, latent-factor model is not a good choice here.

Besides K-means, linear regression, and random forests we use in this assignment, the most advanced method is artificial neural networks(ANNs). It is the most state-of-the-art method right now in machine learning, since it gives the highest accuracy. The idea of ANNs is inspired by the biological neural networks, from animal brains. It learns from the given examples without any specific tasks. One specialized technique in financing (price prediction) is called Large memory storage and retrieval neural networks network[3], a fast deep learning neural networks which can use many kinds of filters at the same time. It analyzes data mainly for the purpose of search and retrieve the hidden information. It automatically selects the key words in data so that each samples in data can be interconnected both horizontally and vertically with appropriate weights. Therefore, it gives the outcome with the best paths in the networks to create the most suitable prediction. It is able to take all features in the dataset and weights features according to their significance when building the interconnected networks, and this is probably why it gives the best prediction.

Because of the limitation of our laptops' functions, we are unable to implement ANNs ourselves. But, the final model we choose, random forest, has some basic common ideas with ANNs model. Random forest model uses effective features to build a decision tree and then search for the optimum prediction based on the branch.

7. REFERENCES

[1] Used cars database
`https://www.kaggle.com/ddmngml/trying-to-predict-used-car-value/data`

[2] How The Random Forests Algorithm Works In Machine Learning
`http://dataaspirant.com/2017/05/22/random-forest-algorithm-machine-learing/`

[3] D. Graupe. *Large memory storage and retrieval (LAMSTAR) network* . April, 1996.