Predicting Prices for Used Cars

Jianqiu Bai
CS Department
College of Wooster
Wooster, OH
jbai18@wooster.edu

Zhirui Zhang
CS Department
College of Wooster
Wooster, OH
zzhang18@wooster.edu

Abstract

This project is on predicting the price of used cars from a data set by using regression trees algorithm. A Python graphics user interface (GUI) is implemented here through Enthought Canopy, in which the program asks the user to select or type the inputs for certain attributes, and then the program predicts and displays the price for the user according to the algorithm mentioned above. This paper outlines the entire processes from predicting the price for used cars to implementing a program for practical use.

Introduction

The used car market is very large in the U.S. If people want to buy a used car, they can find many websites where sellers put the car information. However, everyone wants a value for their assets. Thus, one can see that a used car price predictor is necessary.

Predicting the value of a used car is not a simple task. The value of used cars depends on a number of factors. The most important factors that people would consider are usually the age of the car, mileage (the number of kilometers it has run), brand, model, the extent of damage, gearbox (transmission), and its horsepower. Due to rising fuel prices, choosing gasoline vehicles or diesel vehicles also becomes a considering factor (Pudaruth 2014).

In this project, we are focusing on eight main factors: the age of the car, mileage, brand, vehicle type, the extent of damage, transmission, engine power, and the type of fuel. Also, we are training the data with the regression trees algorithm.

Background

Before starting our own project, we found two related journal articles as particular information: "Predicting the Price of Used Cars using Machine Learning Techniques" and "Predicting the Price of Second-hand Cars using Artificial Neural Networks".

In the first journal article, the author briefly explains the data pre-processing and the implementation of four techniques for the prediction: multiple linear regression analysis method, $K-$nearest neighbours, decision trees and Naive Bayes algorithm. By comparing and evaluating the results gained from different methods, this journal article motivates us to find efficient methods for price prediction, and guides us on how to construct our project (Pudaruth 2014).

The second journal article mainly focuses on using a neural network to predict the price of used cars. The prime attributes in this case are year, make, engine, paint type, transmission and mileage. From this research, the authors found out that the Neural Network provides minimal errors compared to other methods such as k-Nearest Neighbour and Linear Regression (Peerun, Chummun, and Pudaruth 2015).

After evaluating these two journal articles, we finally decided to use the regression trees algorithm as explained in the following sections.

Problem Description

Recall that the goal of this project is making a used car price predictor and analyzing the algorithms involved. Thus, the problems of this project are:

1. Searching the proper data and processing the data using the $numpy$ and $pandas$ Python library extensions.

2. Implementing the learning algorithm by $sklearn$, a Python machine learning library.

3. Analyzing the errors.

4. Implementing the GUI in Python using the $TKinter$ library.

Data Pre-processing

Attributes Selection

The data set contains over 370,000 used cars' information obtained from Ebay Kleinanzeigen (KaggleUser 2016). In this raw data set (see figure 1), there are 19 attributes with another target column which is the price. As not all attributes are useful for price prediction, we only use nine of the ones mentioned above.

	dateCrawled	name	seller	offerType
0	2016-03-24 11:52:17	Golf_3_1.6	privat	Angebot
1	2016-03-24 10:58:45	A5_Sportback_2.7_Tdi	privat	Angebot
2	2016-03-14 12:52:21	Jeep_Grand_Cherokee_"Overland"	privat	Angebot
3	2016-03-17 16:54:04	GOLF_4_1_4__3T`RER	privat	Angebot
4	2016-03-31 17:25:20	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot

	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model
0	480	test	NaN	1993	manuell	0	golf
1	18300	test	coupe	2011	manuell	190	NaN
2	9800	test	suv	2004	automatik	163	grand
3	1500	test	kleinwagen	2001	manuell	75	golf
4	3600	test	kleinwagen	2008	manuell	69	fabia

	kilometer	monthOfRegistration	fuelType	brand	notRepairedDamage
0	150000	0	benzin	volkswagen	NaN
1	125000	5	diesel	audi	ja
2	125000	8	diesel	jeep	NaN
3	150000	6	benzin	volkswagen	nein
4	90000	7	diesel	skoda	nein

	dateCreated	nrOfPictures	postalCode	lastSeen
0	2016-03-24 00:00:00	0	70435	2016-04-07 03:16:57
1	2016-03-24 00:00:00	0	66954	2016-04-07 01:46:50
2	2016-03-14 00:00:00	0	90480	2016-04-05 12:47:46
3	2016-03-17 00:00:00	0	91074	2016-03-17 17:40:17
4	2016-03-31 00:00:00	0	60437	2016-04-06 10:17:21

Figure 1: The sample raw data set for the first five rows.

Removing Noisy Data

In the raw data, several rows contain a price over $\$1,000,000$, which does not really make sense for a used car. Additionally, some prices even go over $\$5,000,000$. Also, there are some rows with missing price which have been set as 0. If the algorithm learns from those examples, the error increase . Thus, our next step is to remove all noisy data. Figure 2 shows the procedure of removing the noisy data.

	price	vehicleType	yearOfRegistration	gearbox	powerPS	model
0	480	NaN	1993	manuell	0	golf
1	18300	coupe	2011	manuell	190	NaN
2	9800	suv	2004	automatik	163	grand
3	1500	kleinwagen	2001	manuell	75	golf
4	3600	kleinwagen	2008	manuell	69	fabia
5	650	limousine	1995	manuell	102	3er
6	2200	cabrio	2004	manuell	109	2_reihe
~~7~~	~~0~~	~~limousine~~	~~1980~~	~~manuell~~	~~50~~	~~andere~~
8	14500	bus	2014	manuell	125	c_max
9	999	kleinwagen	1998	manuell	101	golf

	kilometer	fuelType	brand	notRepairedDamage
0	150000	benzin	volkswagen	NaN
1	125000	diesel	audi	ja
2	125000	diesel	jeep	NaN
3	150000	benzin	volkswagen	nein
4	90000	diesel	skoda	nein
5	150000	benzin	bmw	ja
6	150000	benzin	peugeot	nein
7	40000	benzin	volkswagen	nein
8	30000	benzin	ford	NaN
9	150000	NaN	volkswagen	NaN

Figure 2: A sample procedure of removing the noisy data.

Missing Values

What is more, the data also contains missing values. According to the text book (Mitchell 1997), the easiest approach to deal with the missing values is to assign the most common value for that attribute, which is shown in figure 3.

	price	vehicleType	yearOfRegistration	gearbox	powerPS	model
0	480	NaN → kleinwagen	1993	manuell	0	golf
1	18300	coupe	2011	manuell	190	NaN
2	9800	suv	2004	automatik	163	grand
3	1500	kleinwagen	2001	manuell	75	golf
4	3600	kleinwagen	2008	manuell	69	fabia

	kilometer	fuelType	brand	notRepairedDamage
0	150000	benzin	volkswagen	ja ← NaN
1	125000	diesel	audi	ja
2	125000	diesel	jeep	NaN
3	150000	benzin	volkswagen	nein
4	90000	diesel	skoda	nein

Figure 3: The procedure of replacing the missing values with the most common values.

Data Encoding

In addition, some of these attributes are recorded as integers while others are recorded as strings. Nevertheless, the learning algorithms we are using don't apply to string parameters, so encoding the data from string type to integer type becomes our next task (see figure 4).

	price	vehicleType	yearOfRegistration	gearbox	powerPS	model
0	480	1	1993	1	75	1
1	18300	2	2011	1	190	2
2	9800	3	2004	2	163	3
3	1500	1	2001	1	75	1
4	3600	1	2008	1	69	4
5	650	4	1995	1	102	5
6	2200	5	2004	1	109	6
8	14500	6	2014	1	125	7
9	999	1	1998	1	101	1
10	2000	4	2004	1	105	8

	kilometer	fuelType	brand	notRepairedDamage
0	150000	1	1	1
1	125000	2	2	1
2	125000	2	3	1
3	150000	1	1	2
4	90000	2	4	2
5	150000	1	5	1
6	150000	1	6	2
8	30000	1	7	1
9	150000	2	1	1
10	150000	1	8	2

Figure 4: A sample encoded data set.

Regression Trees Algorithm

After pre-processing the raw data set, we are using the regression trees algorithm to train this data set and predict the price.

The regression trees and the classification trees are two major types in the decision tree learning algorithm. For both

types, each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label (Rahul Saxena 2017b). The classification trees compute the information gain to find the internal node, while the regression trees compute the squared error (Cosma Shalizi 2006). For classification preference, one can use classification trees. For prediction, we are using regression trees instead of classification trees.

The regression tree algorithm places the best attributes of the data set at the root of the tree, then splits the training set into subsets. Subsets should be made in such a way that each subset contains data that has the same attribute value, and finally repeats step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree (Rahul Saxena 2017a).

Error Analysis

We train the first 80% of our data set by using the regression trees algorithm, and predict the price for the last 20% of our data set. By comparing the predicted prices with the actual prices of the last 20% data, we obtain the error graph shown below.

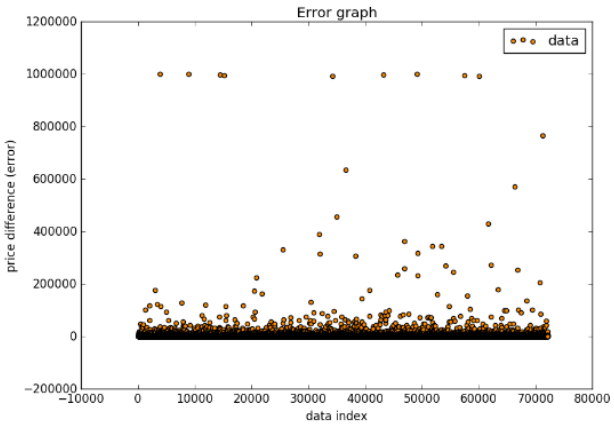

Figure 5: The price difference between the predicted prices and the actual prices for the last 20% of our data set.

According to figure 5, one can see that the price differences for most data are relatively small (around 0). Based on common sense, we set our error tolerance as $3,000$, which means that we can stand our predicted price more or less within the range of $3,000$ to the actual price. The last 20% data set contains 72136 data samples, and there are 8270 data samples having the price difference greater than $3,000$. Thus, the accuracy of our results can be simply computed as 88.54%.

Noting that the price difference for some data sets are extremely large (like $1,000,000$). The reason is that, after we remove the noisy data, there are still some data sets containing prices that are slightly smaller than $1,000,000$ that has been trained by the regression trees algorithm, which increases the error a little bit.

GUI Implementation

Having the data managed and the algorithm trained, we can finally implement a Graphics User Interface (GUI), which reads the regression trees' nodes obtained during training. Then the program will use the generated rules to predict the price every time the user types new inputs. A sample GUI is shown in Figure 6.

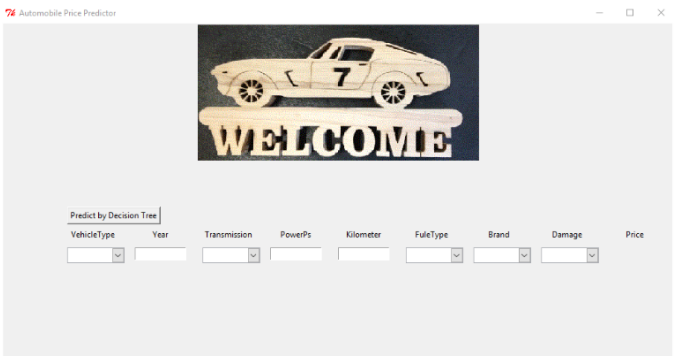

Figure 6: The GUI for predicting used-car prices.

Conclusion

This paper states the entire procedure of predicting the prices for used cars. The paper first introduces the procedure of pre-proccessing the raw data. Then the regression trees algorithm is explained. Next, the error analysis is followed to illustrate the performance of the regression trees. In the end,a GUI is implemented for practical use.

One limitation is that we manually set our error tolerance as $3,000$. As we experimental reduce the error tolerance to $2,000$ and 1000, the accuracy reduces to around 82.26% and 64.59%. Thus, the accuracy in our project reduces sensitively as the error tolerance reduces.

The other limitation is that we only implement the regression trees algorithm. In the future, we hope to predict the prices with more learning algorithms, so we can compare the different algorithms and use one of them that has the best performance.

References

Cosma Shalizi. 2006. Regression Trees. http://www.stat.cmu.edu/ cshalizi/350-2006/lecture-10.pdf. [Online; accessed 5/7/2017].

KaggleUser. 2016. Used cars database. https://www.kaggle.com/orgesleka/used-cars-database. [Online; accessed 4/18/2017].

Mitchell, T. M. 1997. *Machine Learning*. McGraw-Hill Science/Engineering/Math.

Peerun, S.; Chummun, N. H.; and Pudaruth, S. 2015. Predicting the Price of Second-hand Cars using Artificial Neural Networks. *Proceedings of the Second International Conference on Data Mining*.

Pudaruth, S. 2014. Predicting the Price of Used Cars using Machine Learning Techniques. *International Journal of Information Computation Technology*.

Rahul Saxena. 2017a. BUILDING DECISION TREE ALGORITHM IN PYTHON WITH SCIKIT LEARN. http://dataaspirant.com/2017/02/01/decision-tree-algorithm-python-with-scikit-learn/. [Online; accessed 5/7/2017].

Rahul Saxena. 2017b. How decision tree algorithm works. http://dataaspirant.com/2017/01/30/how-decision-tree-algorithm-works/. [Online; accessed 4/18/2017].