Prediction of the Advertised Price of a Used Car

Capstone Project for Udacity Machine Learning Engineer Nanodgree

Rui Cao

September 9, 2017

1 Definition

1.1 Project Overview

In the used car market, one important problem is to determine a used car's price. A seller always tries 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.

1.2 Problem Statement

This 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 these two quantities, i.e. the absolute 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.

1.3 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 may be evaluated using mean absolute error between the two prices, which are:

$$E_{ae} = \frac{\sum_{i=1}^{N} |P_p - P_a|}{N}$$

where E_{ae} is the mean absolute error, P_p is the prediction of the car's price, P_a is the car price advertised.

2 Analysis

2.1 Data Exploration

The dataset is extracted from a used car database on Kaggle. It consists 371528 rows of data. Each row contains following information of a used car advertisement, note that in some fields there may be empty value:

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 An 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.

The dataset is read using pandas.read_csv from a csv file, all fields without information in it is filled with an empty string. 7 samples in the dataset are shown as below:

Figure 1: First Four Samples in the Dataset

	0	1	2	3
dateCrawled	2016-03-24 11:52:17	2016-03-24 10:58:45	2016-03-14 12:52:21	2016-03-17 16:54:04
name	Golf_3_1.6	A5_Sportback_2.7_Tdi	Jeep_Grand_Cherokee_"Overland"	GOLF_4_1_4__3T�RER
seller	privat	privat	privat	privat
offerType	Angebot	Angebot	Angebot	Angebot
price	480	18300	9800	1500
abtest	test	test	test	test
vehicleType		coupe	suv	kleinwagen
yearOfRegistration	1993	2011	2004	2001
gearbox	manuell	manuell	automatik	manuell
powerPS	0	190	163	75
model	golf		grand	golf
kilometer	150000	125000	125000	150000
monthOfRegistration	0	5	8	6
fuelType	benzin	diesel	diesel	benzin
brand	volkswagen	audi	jeep	volkswagen
notRepairedDamage		ja		nein
dateCreated	2016-03-24 00:00:00	2016-03-24 00:00:00	2016-03-14 00:00:00	2016-03-17 00:00:00
nrOfPictures	0	0	0	0
postalCode	70435	66954	90480	91074
lastSeen	2016-04-07 03:16:57	2016-04-07 01:46:50	2016-04-05 12:47:46	2016-03-17 17:40:17

Figure 2: Last Three Samples in the Dataset

	371525	371526	371527
dateCrawled	2016-03-19 18:57:12	2016-03-20 19:41:08	2016-03-07 19:39:19
name	Volkswagen_Multivan_T4_TDI_7DC_UY2	VW_Golf_Kombi_1_9l_TDI	BMW_M135i_vollausgestattet_NP_52.720___Euro
seller	privat	privat	privat
offerType	Angebot	Angebot	Angebot
price	9200	3400	28990
abtest	test	test	control
vehicleType	bus	kombi	limousine
yearOfRegistration	1996	2002	2013
gearbox	manuell	manuell	manuell
powerPS	102	100	320
model	transporter	golf	m_reihe
kilometer	150000	150000	50000
monthOfRegistration	3	6	8
fuelType	diesel	diesel	benzin
brand	volkswagen	volkswagen	bmw
notRepairedDamage	nein		nein
dateCreated	2016-03-19 00:00:00	2016-03-20 00:00:00	2016-03-07 00:00:00
nrOfPictures	0	0	0
postalCode	87439	40764	73326
lastSeen	2016-04-07 07:15:26	2016-03-24 12:45:21	2016-03-22 03:17:10

First observation provides following conclusions:

1. It's hard to tell the meaning of the 'abtest' field. So it's difficult to explore the relation between this field and price determination.

2. The field 'name' basically contains the information of a car's brand and model, so that field is closed related to the fields of 'brand' and 'model'.

3. The fields 'seller', 'offerType' and 'nrOfPictures' are possiblly having the same value for all records. If that is true, then those fields are useless for price determination.

4. The fields 'vehicleType', 'model', 'gearbox', 'feulType' and 'notRepairedDamage' are having some records without value in it, i.e., having an empty string as the value. So all potential useful fields will be checked to see if there are the records having empty values in the field.

5. There are unreasonable values in 'powerPS' and 'monthOfRegistration' fields. They are supposed to be non-zero. So all numerical fields will be checked to see if there are unreasonable values.

6. Though 'monthOfRegistration' and 'postalCode' appear in numerical forms, they don't have specific quantitative meanings with respect to the price determination. So they should be treated the same as text fields.

7. The date fields, 'dateCrawled', 'dateCreated' and 'lastSeen'appear in text forms. But they can be transformed into numerical forms by calculating the number of days between the value and a fixed level date.

Therefore some further explorations are taken. Values are grouped in the fields 'seller', 'offerType', and 'nrOfPictures' respectively, and the distributions are shown below:
Distribution of data in the 'seller' field:

```
gewerblich          3
privat         371525
```

Distribution of data in the 'offerType' field:

```
Angebot    371516
Gesuch          12
```

Distribution of data in the 'nrOfPictures' field:

```
0    371528
```

As can be seen, all records have the same value in 'nrOfPictures' field. So this feature is useless for price determination and will be dropped from the dataset. While too few records are having value 'Gesuch' in the 'offerType' field, or value 'gewerblich' in the 'seller' field. So those records won't be considered in this problem. Then the two fields will be dropped from the dataset.

For all the other text fields, number of records with empty values in those fields are checked, the results are shown below:

Table 1: Number of Records with Empty Value in the Field

field name	number of empty records
dateCrawled	0
name	0
abtest	0
vehicleType	37869
gearbox	20209
model	20484
fuelType	33386
brand	0
notRepairedDamage	72060
dateCreated	0
lastSeen	0

As mentioned before, missing value in the 'model' field may be repaired by the information extracted from the 'name' field. While for missing values in other fields 'vehicleType', 'gearbox', 'fuelType' and 'notRepairedDamage', there are no obvious way to repair them using existed information. So those records with empty values in these four fields will be discarded.

Some statistical information are presented below for the numerical fields:

Figure 3: Statistics of Numerical Fields

	price	yearOfRegistration	powerPS	kilometer	monthOfRegistration	postalCode
count	3.715280e+05	371528.000000	371528.000000	371528.000000	371528.000000	371528.00000
mean	1.729514e+04	2004.577997	115.549477	125618.688228	5.734445	50820.66764
std	3.587954e+06	92.866598	192.139578	40112.337051	3.712412	25799.08247
min	0.000000e+00	1000.000000	0.000000	5000.000000	0.000000	1067.00000
25%	1.150000e+03	1999.000000	70.000000	125000.000000	3.000000	30459.00000
50%	2.950000e+03	2003.000000	105.000000	150000.000000	6.000000	49610.00000
75%	7.200000e+03	2008.000000	150.000000	150000.000000	9.000000	71546.00000
max	2.147484e+09	9999.000000	20000.000000	150000.000000	12.000000	99998.00000

Further observations about the numerical fields:

1. Unreasonable 0 values appear in the field 'price', 'yearOfRegistration', 'powerPS', and 'monthOfRegistration'.

2. Unreasonable early and late years, such as '1000' and '9999', appear in the field 'yearOfRegistration'.

3. Most of the price data fall into range of thousands of dollars, while there are unreasonable high prices, such as 2.1e9.

4. There are unreasonable high values, such as '20000', appear in the field 'powerPS'.

5. More than half of the used cars advertised are having value 150000 in the field 'kilometer'. This indicated that data in the field kilometer may be inaccurate. It is highly possible that in ebay, the largest possible value that can be filled for a used car's kilometer is 150000.

Based on the above observations, records with unreasonable values will be discarded. A common sense provides the ranges of values in fields of 'yearOfRegistration' and 'powerPS' to classify the reasonable values. 'yearOfRegistration' should be within the interval of [1950,2016] to make sure the car advertised is a used car, instead of a new car, or an antique car. The number of 'powerPS' should be less than 1500, the known highest powerPS of a car.

It's obvious that the price data have many outliers, so the top 0.1% prices are cut off. The remain records have a more reasonable maximum value and should have much less outliers than before.

2.2 Exploratory Visualization

For records with reasonable values, make scatter plots of 'price' vs 'yearofRegistration', 'powerPS', and 'kilometer' respectively. From the plots below, it can be seen that, there are no obvious linear relations between the numerical features and the price data. So a simple linear model is less likely to make precise price predictions for the used cars in this dataset.

Figure 4: Price vs Year of Registration

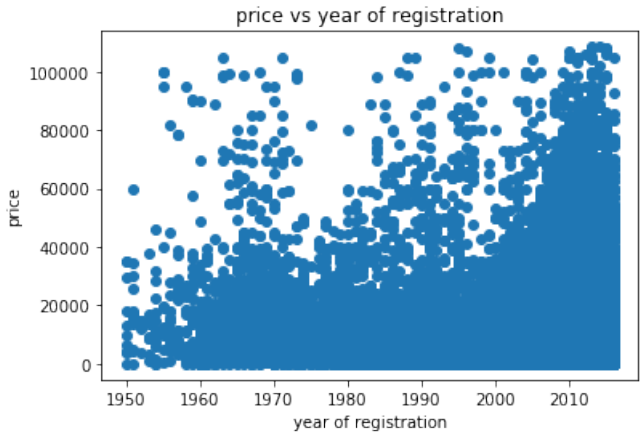


Figure 5: Price vs powerPS

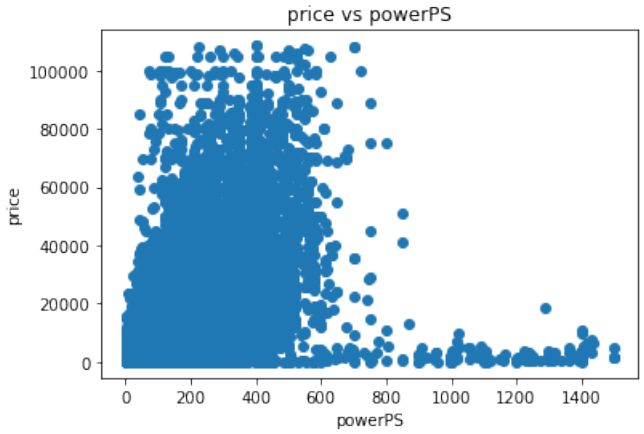
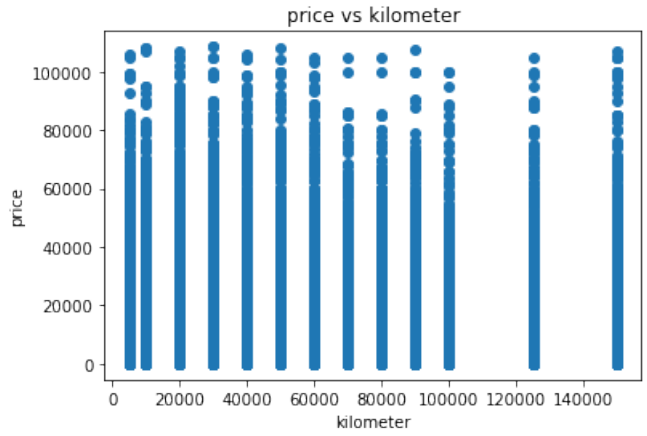


Figure 6: Price vs Kilometer



2.3 Algorithms and Techniques

To this problem, a multilayer neural network model will trained based on the processed data from the dataset. A multilayer neural network properly deals with the following characteristics:

- Large size of dataset. A well chosen multilayer neural network is complicated enough to extract desired information from a large size of data.
- Nonlinear relation between input features and the target variable. With activation functions added within each layer, a multilayer neural network is able to simulate the nonlinear relation between the inputs and outputs.

The original dataset will be pre-processes to generate the input features and price data will be taken as the target variable.

Once the inputs features and price data are ready, they will be split into train set, validation set and test set in a ratio of 6:2:2 by randomly picking from the processed dataset.

Before training a multilayer model, a simple mean value prediction model will be constructed as the benchmark. Then a neural network without inner layer and activation functions will be trained to see how a linear model performs on this problem. A single layer neural network with activation function will be tested to see if a nonlinear model improves the performance. Then a final model will be trained and

its hyper-parameters will be tuned.

There are several hyper-parameters need to be tuned in the model, number of layers, number of nodes in each layer, type of activation functions, and dropout rate. All the hyper-parameters will be tuned by grid searching in a specific set.

2.4 Benchmark

First a naive predictor will be created as the benchmark. The predictor simply uses the mean value of all prices in train set as its prediction for an unseen car, i.e.:

$$P_n = \frac{\sum_{i=1}^{N_0} P_{t,i}}{N_0}$$

where P_n is the naive prediction of the price, N_0 is the number of samples in train set, $P_{t,i}$ is the i-th price data in the train set.

3 Methodology

3.1 Data Pre-Processing

The original dataset will be processes as following:

1. 'nrOfPictures' field is dropped because all values in this field are 0.
2. Records with minority value in 'seller' or 'offerType' fields will be discarded. Then the two fields are dropped.
3. Records with empty value in 'gearbox', 'vehicleType', 'fuelType' or 'notRepairedDamage' fields are discarded.
4. Records with empty value in 'model' field are repaired by extracting information from 'name' field. Then records not fixed are discarded and 'name' field is dropped. Strings in 'brand' and 'model' fields will be turned into upper case to eliminate duplications with different cases.
5. Records with unreasonable value in 'price', 'powerPS', 'yearOfRegistration', or 'monthOfRegistration' fields are discarded. Then records with top 0.1% prices are discarded.

6. 'dateCrawled', 'dateCreated' and 'lastSeen' fields are transformed into numerical forms, by first converting string values in the fields into datetime data, then the datetime data are transformed into numerical data by calculating the days between the date in the data and a fixed level date.

7. Values in 'postalCode' field will be cut off the last three digits, then the remained one or two digits will be treated as text.

8. 'price' field will be extracted as the target variable data.

9. 'abtest', 'vehicleType', 'gearbox', 'model', 'fuelType', 'brand', 'notRepaired-Damage', 'monthOfRegistration' and processed 'postalCode' fields will be treated as text fields. One-hot encoding will be applied to generate input features from the values in those fields.

10. 'powerPS', 'yearOfRegistration' and 'kilometer' fields will be treated as numerical fields, and normalized by a min-max scaler.he car's price, P_a is the car price advertised.

Finally, the processed data have 244588 records which can be used for model training and testing. These records are split into train set, validation set and test set by randomly picking and the sizes of the sets are in a ratio of 6:2:2. Then train set has 146752 samples. Both of validation set and test set have 48918 samples respectively.

3.2 Implementation

First implement the naive model as a benchmark. It's seen that the naive predictor get a score of approximately 5439.84, meaning that averagely the difference between the prediction price and the true price is about $5439.84. So naive predictor is not performing very well. Next, a more effective model will be created, evaluated and compared with this benchmark model.

The Keras package is used to implement all the neural network models. A linear model is first trained to see if a neural network model works for this problem.

The linear model consists only one inner layer with 128 nodes and linear activation function. So the total parameters in the model are 54529. The optimizer is Adam and the loss function is the mean absolute error function. The model is trained with 20 epochs and with batch size of 500. It gets a score of 3569.43 for the

test set.

So the linear model provides much better prediction prices comparing with the naive model, indicating a neural network model works for this problem. However, without future processing of the features, a linear model is still not good enough to solve the problem. While properly handling with nonlinear relation between features is difficult and needs many experiences and experiments. A more convenient way is to employ a multilayer neural network with nonlinear activation functions. Then the model will automatically find out the best way to represent the nonlinear relations between features and the price.

Therefore, a neural network with two inner layers and nonlinear activation functions are trained and tested. The model has 128 nodes in the first inner layer and 32 nodes in the second inner layer, each layer has relu function as the activation function. The model has 58561 parameters to be trained. The optimizer is Adam and loss function is mean absolute error function. The model is trained with 20 epochs and with batch size of 500. It gets a score of 1784.62 for the test set.

A simple single layer neural network with relu activation function improves the prediction prices with the error almost halved. This indicates that introducing more nonlinear properties of the model indeed improves its performance. So next a multilayer neural network will be created and its hyper-parameters will be carefully tuned to find the best model for solving the problem.

3.3 Refinement

To tune the hyper-parameters of the multilayer neural network, following experiments have been taken:

1. 2,3 or 4 layers are added to the model to determine how many layers gives the best score.

2. Different number of nodes in each layer are tested within the set of {1024, 512, 256,128, 64, 32, 16, 8} to determine how many nodes should each layer has.

3. Type of activation functions are tuned, in the set of {'relu', 'sigmoid', 'softmax', 'tanh'}.

4. Dropout rate within each layer are also tuned. Dropout rate in the set of {0 ,0.2, 0.4, 0.5} are tested.

Finally, the model provides an acceptable outcome has the following four inner layers:

1. First inner layer with 128 nodes ,relu activation function, and no dropout.
2. Second inner layer with 64 nodes, linear activation function, and no dropout.
3. Third inner layer with 32 nodes, relu activation function, and no dropout.
4. Fourth inner layer with 8 nodes, linear activation function, and no dropout.

The model has 65009 parameters to be trained. The optimizer is Adam and loss function is mean absolute error function. The model is trained with 20 epochs and with batch size of 500. This final model gets a score of 1361.26 for the test set.

Note that the scores for all those neural network models may vary within a very small range for each run of the code.

4 Results

4.1 Model Evaluation and Validation

The final model has four inner layers with reasonable number of nodes in each layer, it takes processed data for a used car ad and predicts the price of the car. Based on its performance on data in the test set, averagely the model's prediction has an approximately $1340 mean absolute error compared to the actual price. This is acceptable, and much better than the prediction made by the naive mean predictor, which is taken as the benchmark and has approximately $5440 mean absolute error compared to the actual price.

The processed dataset is re-split into train set, validation set and test set randomly with a different random seed to test the model's robust. The re-trained model gets a score of 1344.16 for the changed test set. Since there is no significant change in the training process and the model's score, the model is considered to be robust to small changes of input data.

4.2 Justification

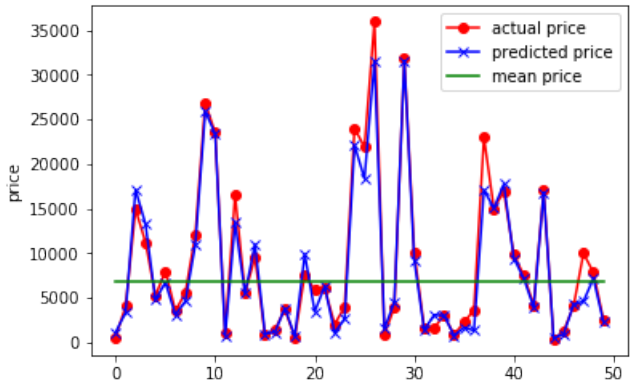
Using the mean absolute error as the metric, the benchmark model gets a score of approximately 5440, while the final model gets a score of approximately 1340. So

the final model makes predictions with much less absolute error compared to the benchmark model.

To justify the outcome of the final model, we randomly pick 50 samples from the test set and compare the actual price and price prediction, for most of the samples, the model makes good predictions, i.e., the predicted price is much closer to the actual price, compared to the naive mean prediction. However, there are some samples having large error between the prediction price and the actual price, especially for those records with actual prices larger than $10000. Two possible factors contributes to this phenomenon:

- More than 78% of the data have prices less than $10000. So train data for cars with prices higher than $10000 is in a small size. The inaccurate prediction to cars with high actual prices may be due to inadequate data for training the model.
- The price of the advertised used car is not the exact value of that car. Because that price is affected by personal factors of an advertiser, so this price is not only determined by the details of a used car, but also related to the advertiser. However, there is no information about the advertiser in the dataset. Therefore, the inaccurate prediction to cars may also be due to inadequate information from the dataset.

Figure 7: Prices Comparison for 50 Randomly Picked Samples



From all the above, when the model's prediction is less than $10000, it can be taken as a relatively accurate prediction. While if the prediction is larger than

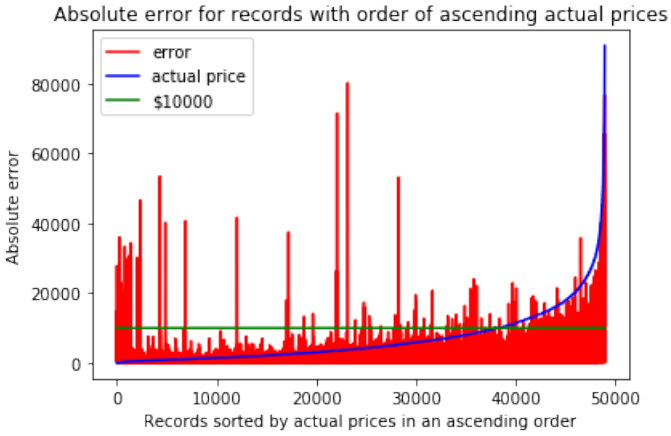
$10000, it can be seen as a reference to indicate approximately in what range of the used car's price is, but may not be accurate enough to directly evaluate the used car.

5 Conclusion

5.1 Free-Form visualization

In the previous section, generally we see that prediction prices are more accurate for cars with low actual prices. We plot the absolute error for records ordered by actual prices to see if that conclusion makes sense. From the plot, it is seen that for most records with actual prices less than $10000, the absolute error keeps low. As long as the actual prices gets higher, the absolute error gets higher as well. Meanwhile, when the actual price is extremely low, for example, less than $500, the absolute error is also in a high level.

Figure 8: Absolute Error with Respect to Ascending Ordered Actual Price



The above observation suggests that the model performs differently for three types of data. It produces high absolute error for cars with very low or very high advertised prices, and makes more accurate predictions for cars with actual prices within a range that is neither too low nor too high. This can be verified by train a model for only records with actual prices within the range of [500,10000].

The model takes the same structure of the final model and gets a score of 811.79 for the test set. So by only using data with prices in a narrow range, the model's

performance is improved. So building a sub-model to determine if the used car is in a normal price class, a low price class or a high price class, then train models differently according to each class will improve the model's performance for the whole dataset.

5.2 Reflection

The process to solve the problem in this project is as following:

1. A proposed problem and its related dataset were found, described, and presented.
2. The dataset was investigated by sample observation, statistical investigation and visualization.
3. The candidate model and metrics were discussed.
4. The dataset was preprocessed and then be transformed into input data for training, validating, and testing the model.
5. A benchmark model was implemented and tested.
6. The neural network model was then implemented and improved from a simple linear model to a complicate nonlinear model.
7. Hyper-parameters of the final model were tuned.
8. Model's outcome were observed and investigated. An experiment was taken to suggest a potential way to further improve the model.

It takes a lot of time in the stage 7 to tune the hyper-parameters. At first, as more as possible layers and nodes are taken, then to reduce the overfitting, less and less layers and nodes are taken. At some point the model performance will decrease due to underfitting, then a finer search will be applied near that point of hyper-parameters to find out the best model.

It also needs a lot of efforts to clean the abnormal records in dataset and preprocess the remained the data. Some ranges have been applied the the numerical fields to filter out the outliers from the data. While those ranges could be adjusted and will definitely affect the model's performance, as shown in the previous subsection.

It should noted that, the dataset itself may have some defects that can not be repaired. By realizing this, the process of collecting precise, accurate, relevant data with comprehensive information is one the most factors for successfully solving a real problem.

5.3 Improvement

As discussed before, a hybrid model may be developed as following to improve the outcome of this project:

1. A classification model is first trained to make a prediction of in what range a used car's price falls. There are three group of the used cars with prices in low range, normal range and high range, for example ranges of (0,500), [500,10000], (10000, 1000000].

2. For each class, a separate model will be trained with respect to the specific group of the data.

3. The two models will be combined to produce a hybrid model for generating predictions for used cars.