INF 552 Final Project - Price Prediction of Used Cars

University of Southern California
Zihao Wang
Shuying Ye
Xiao Chen

First of all, within the zip file we handed in, there are four Python files: "data_pre_processing.py", "prepare_data_for_LR.py", "prepare_data_for_tree.py", and the main file for running the program called "main_file.py". The dataset we used for training is in the file "used_car_train_20200313.csv", and the dataset we used to test our program in the file "used_car_testA_20200313.csv". The file "prdict_result.csv" stores the final prediction result of our program. In this project, each member has put equal effort into this project.

1. Introduction

This project aims to predict the selling price of used cars. The project utilized four different models in performing this price prediction – Linear Regression, Neural Network, Decision Tree, and Random Forest. Each of the algorithm ran separately, where the programs were first trained by using the training dataset *"used_car_train_20200313.csv"*, then tested by using the testing dataset *"used_car_testA_20200313.csv"*. *K-fold Cross-Validation* (Brownlee, 2018) was then introduced to evaluate the performance of each of the models, in which models were evaluated based on Mean Average Error (MAE). A smaller MAE is more preferable – some models would be dropped if it has an unacceptable MAE that is higher than one, while others with acceptable MAE would be kept as satisfied models. Lastly, weights of the satisfied models – meaning that how influential the result of this model would be, were calculated based on the different scores (the MAE) of the satisfied models. The final prediction would be an integration of the results from all satisfied models, calculated by using their corresponding results and weights.

2. The Four Algorithms

Linear Regression

The linear regression algorithm performs a regression task to model a target prediction value based on the independent variables. It is most commonly used to figure out the relationship between variables and make a forecast based on the input variables.

The advantages of Linear Regression algorithm include, for example, ideal for capturing the linear relationship among concentrated data. Besides, it offers a high training and forecasting speed. When there is newly added data, Linear Regression offers an easy update of the model. Moreover, the algorithm tends to have a good performance when dealing with small datasets.

However, the Linear Regression algorithm also has some disadvantages. Firstly, it would not be appropriate for nonlinear data. Also, the ability for the algorithm to separate signals and noise is weak. It is also thought to be having a relatively low accuracy when making predictions.

Neural Network

The neural network algorithm is a system that learn to perform tasks without being programmed with specific rules. It is inspired by the biological neural networks of animal brains (Chen et al., 2019). This algorithm aims to recognize patterns, it uses machine perception, clustering, or labeling to manage the raw input, so as to interpret sensory data (Kumar Thittamaranahalli, 2020). The algorithm could help us perform data clustering and categorizing.

Based on the similarity and pattern of the input training data, the algorithm is able to categories new data which is not yet labelled.

Neural network could be used to perform prediction task. By establishing correlations between present and future events, it is able to run regression between the past and the future, hence being able to get a prediction about the future.

However, there are also some limitations of the neural network algorithm. Firstly, the "black box" characteristic of Neural Network – the process of how and why certain output is gained would not be known, thus it is lack of interpretability. Besides, the algorithm tend to require much more amount of data for training compare to other traditional Machine Learning algorithms. Also, Neural Network is usually more computationally expensive than traditional algorithms.

Decision Tree

One the other hand, while neural network is lack of interpretability, the Decision Tree algorithm is very interpretable. The ID3 algorithm generates decision trees from a dataset. For example, if we were interested in investigating which factors would determine a student's GPA. By implementing ID3 Algorithm, based on datasets such as SAT scores and Average hours of studying, we would be able to train the prediction model (Decision tree) to predict a student's GPA.

There are some advantages of the Decision Tree algorithm. To begin with, the Decisions Tree model is very intuitive. Secondly, Decision Tree can be visualized, and tree is easy to interpret. Moreover, the Decision Tree dataset does not require normalization.

On the other hand, this algorithm also have some disadvantages. First of all, the ID3 does not assure an optimal solution. It is based on choosing the most appropriate attribute to split the dataset on every iteration. Also, the algorithm is based on Occam's razor, which means "the simplest solution is mostly likely to be the right one". On the whole, Occam razor is intuitively reasonable, but it does not provide solid mathematical proofs like other algorithm does (e.g. Regression). Furthermore, the ID3 can overfit to the guided facts, to avoid overfitting, we should make use of smaller trees instead of using larger ones (Batra and Agrawal, 2018). Meanwhile, in cases of a extreme large variable size, training Decision Tree with ID3 algorithm could be very costly in terms of computation capacity, and at the risk of overfitting.

Random Forest

To overcome the risk of inaccurate and improve accuracy performed by Decision Tree, we can implement the Random Forest algorithm, which would be seen as an improved version of the ID3 Algorithm. The algorithm simply takes random samples with replacement from the population. (Bootstrap aggregating/Bagging). After building multiple random training subsets, the algorithm constructs one tree per random training subsets. This technique is called the random split selection method and the trees are known as random trees (Sathyadevan and Nair,

2015). Meanwhile, this algorithm creates several trees, and the best estimation would be selected from the majority class value.

3. Methods and Results

Class methods were used in this project. In detailed, four steps were involved in the project, which were data analysis, feature engineering, models selection, and stacking (combining all the learning models).

Step 1 - Data Analysis

The training dataset that contains multiple features are shown as below:

```
['SaleID', 'name', 'regDate', 'model', 'brand', 'bodyType', 'fuelType', 'gearbox', 'power', 'kilometer',
'notRepairedDamage', 'regionCode', 'seller', 'offerType', 'creatDate', 'price', 'v_0', 'v_1', 'v_2', 'v_3','v_4', 'v_5',  'v_6',
'v_7', 'v_8', 'v_9', 'v_10', 'v_11', 'v_12', 'v_13', 'v_14'].
```

Figure 1: Train data set features

The distribution of each of the feature within the dataset was observed. Firstly, the distribution of "price" was shown below:

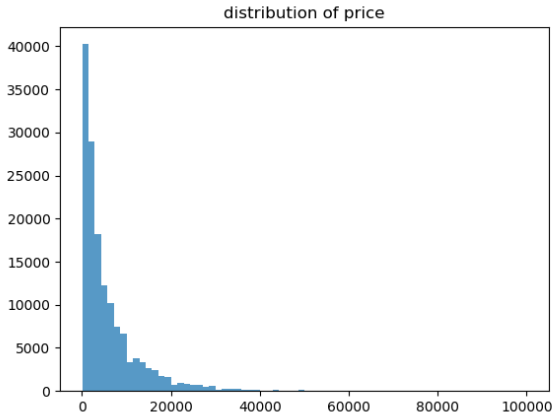

Figure 2: The original distribution of "price"

As shown in Figure 2, the original distribution of "price" does not follow a normal distribution pattern, thus it has to be transformed to achieve a better prediction result. In order to acquire a normal distribution, log transformation was used, and the transformed "price" distribution is shown as figure 3.

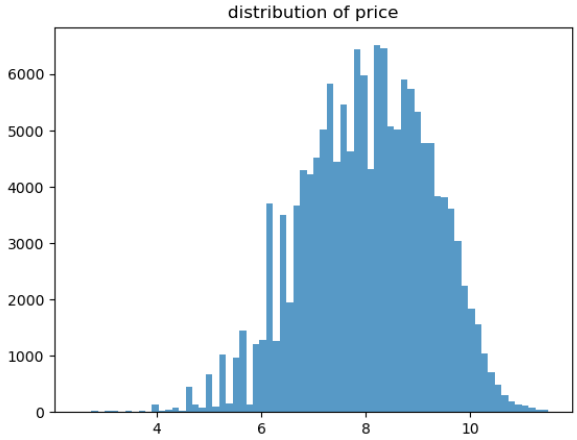

Figure 3: Price distribution after log transform

Secondly, the distribution of NaN values over entire dataset was checked and the result is shown as the following graph (see Figure 4).

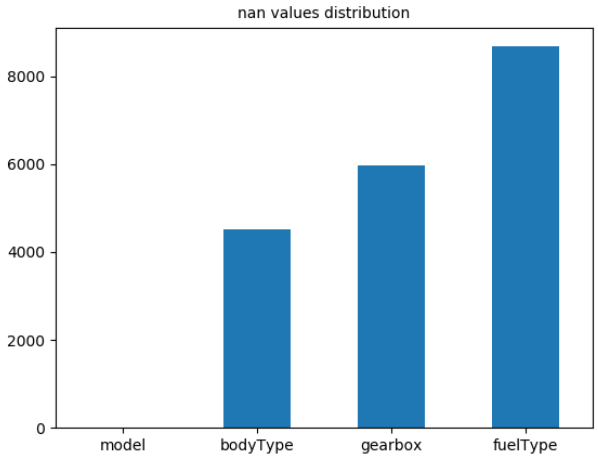

Figure 4: NaN values distribution

To achieve a more accurate predict result, the NaN values of “fuelType”, “gearbox” and “fuelType” was filled with 0s. This is done for two reasons: Firstly, NaN values occupy a large portion in these three features. Secondly, these features are classified features which implies that they have only 0s or 1s.

Step 2 - Feature Engineering

After basic data analysis is finished. Some modifications of features have to be done. To be more specific, after observing each of the features, the feature “power” has to be filtered since it contains some abnormal values. In order to do this, the box-plot (Khan, 2020) method was used and the outputs are shown as below:

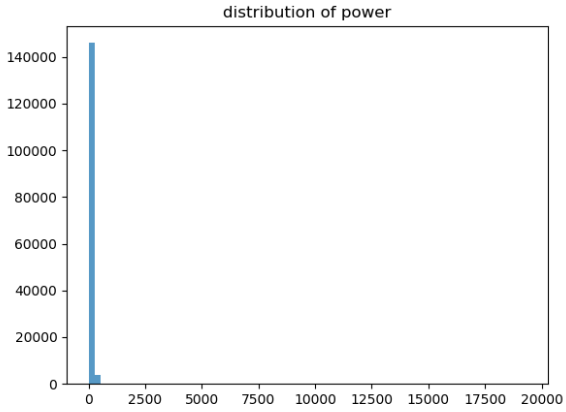

Figure 5: power distribution before filter

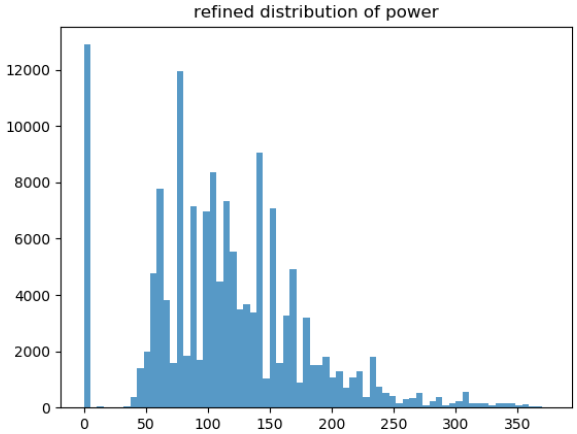

Figure 6: Power distribution after filter

On the other hand, some new features were introduced to refine the train data set. The feature called “age” was introduced which represents the age of used cars. The way in calculating this “age” was straightforward, which was subtracting between “regDate” and “createDate”. Additionally, the median value and mean value of car prices in terms of brands were added to the dataset. Eventually, unneeded features were picked out and dropped, which is done by the way shown in the following figure (see Figure 7):

```
data = data.drop(['SaleID','name','regDate','creatDate','offerType','seller'], axis=1)
```

Figure 7: Drop unneeded features

As four models were used in this project, separate datasets were required to satisfy different requirements of models. To be more specific, numerical features must be normalized to fulfill the requirement of Liner Regression and Neural Network. In addition, classified features with over two cases must be encoded for the same reason.

Step 3 - Models selection

The model selection process was intuitive. In this project, as it is a prediction task, four models including Neural Network, Linear Regression, Decision Tree and Random Forest were chosen.

In analyzing the performance for each model. The concept "K-fold Cross-Validation" (Brownlee, 2018) was introduced, which includes the following steps:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
    1. Take the group as a hold out or test data set
    2. Take the remaining groups as a training data set
    3. Fit a model on the training set and evaluate it on the test set
    4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Figure 8: General steps for K-fold Cross-Validation (Brownlee, 2018)

The MAE (Mean Average Error) was used for the score of each model, which is the average value of mean square errors of all k-folds for each model. In addition, the K value for this project was chosen as five. The reason for choosing five is that, as there is no "best k values", common cases were selected, which is five or ten.

The benefits of this method are obvious. First of all, it prevents models being under-fit or over-fit since it chose subset randomly. Moreover, with this method, it is possible for us to evaluate performances of several models to choose the best one or combine them.

Step 4 - Stacking

Stacking stands for combining results of different models together to get a better result. In this project, the way of stacking is straightforward – it is a weighted combination, in which higher weights were given to the models with better scores.

Results

The scores (the MAEs) of our four models are as the following:

Model	MAE
Random Forest	0.13310130551940452
Decision Tree	0.19090345013355864
Linear Regression	1610.8877575333609
Neural Network	0.19606412441957888

As we can see from the scores, the MAE for the three models: random forest, decision tree, and neural network are small enough (less than one) to be acceptable. However, the score for linear regression, about 1610.89 was far higher than 1, thus it is far exceeding our acceptable range and it was dropped.

Therefore, the final price prediction in this project would be based on combing the weighted results of random forest, decision tree, and neural network, which was then output into the file "predict_result.csv". Some examples of the final prediction are as the following (see Figure 9):

predict_result

	price
0	42229.20635428710
1	298.10685146837100
2	7241.52373607013
3	12505.898534802400
4	549.5301220052360
5	1955.7305797537700
6	5124.225216641190
7	9082.642099069130
8	2516.4906833937500

Figure 9. Examples of Final Price Prediction

4. Limitations

For future study, the project could be further enhanced by using algorithms to adjust the parameters in order to optimize our models. For Decision Tree and Random Forest, there could be a limit set for the maximum number of levels, and this limit could be set by adjusting the parameters. Similarly, for Neural Network, adjusting the parameters could determine the maximum number of layers, and the maximum number of neurons in each layer.

5. Conclusion

After evaluation, for the four algorithms selected in this project, Decision Tree, Random Forest, and Neural Network are ideal for accurately predicting the price of used cars. On the other hand, Linear Regression gave a high Mean Average Error, hence could not be used for the prediction. Further enhancement is possible for this project, and the logic behind this project allows it to also be able to apply to other similar projects where a prediction needs to be made based on some existing feature values.

Reference

Batra M., Agrawal R. (2018) Comparative Analysis of Decision Tree Algorithms. In: Panigrahi B., Hoda M., Sharma V., Goel S. (eds) Nature Inspired Computing. Advances in Intelligent Systems and Computing, vol 652. Springer, Singapore

Brownlee, J. (2018). A gentle introduction to k-fold cross-validation. Accessed October, 7, 2018.

Chen, Y. Y., Lin, Y. H., Kung, C. C., Chung, M. H., & Yen, I. (2019). Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. *Sensors*, *19*(9), 2047.

Khan. (2020). *Summarizing quantitative data | Statistics and probability | Khan Academy.* Khan Academy. Retrieved 10 May 2020, from https://www.khanacademy.org/math/statistics-probability/summarizing-quantitative-data#box-whisker-plots.

Kumar Thittamaranahalli, S. (2020). *Neural Network*. Lecture, University of Southern California.

Sathyadevan S., Nair R.R. (2015) Comparative Analysis of Decision Tree Algorithms: ID3, C4.5 and Random Forest. In: Jain L., Behera H., Mandal J., Mohapatra D. (eds) Computational Intelligence in Data Mining - Volume 1. Smart Innovation, Systems and Technologies, vol 31. Springer, New Delhi