Car Sales Prediction

Mingxin CHENG
20387442
mchengaa@connect.ust.hk

Zhengbo SU
20406963
zsuae@connect.ust.hk

Ziwei WANG
20402072
zwangcp@connect.ust.hk

Zihuan XU
20394639
zxuav@connect.ust.hk

ABSTRACT

Big data is a heated topic in recent years. In this report, we will discuss how we use methods of big data to build a model to predict the sales volume and revenue for each type of car given car sales data and reviews data. We chose three different types of models – Linear Regression, Naïve Bayes and Multilayer Perceptron Regression. Linear Regression is a linear model. Naïve Bayes is a model based on probability. And the Multilayer Perceptron Regression is a neural network model. We predict the sales volume by each model and evaluate the performance of each one.

Keywords

Car Sales Prediction, Data Analysis, Linear Regression, Bayes Regression, Multilayer Perceptron Regression

1. INTRODUCTION

Over these years, big data has become more and more commonly used to improve people's life. Prediction is a very important usage of big data. In this article, we use several regression models to predict the future sales of cars.

The article is structured as follows: section 1 introduce the background and problem to be solved, as well as the related works has been done. In section 2, we will overview the whole project and go through all the steps. You can get a rough idea of what things are done in this section. Section 3 describes how we processed the data, including crawling new data, preprocessing the source data, and extracting features from the preprocessed data. Section 4 provides an introduction of all the models we used in this project: linear, Bayesian, MLP and combined model. The section 5 displays an evaluation of all the models we used, compares their performance. The article concludes in Section 6 with a discussion of the obtained results and the lessons we learned in this project. This section also provides thoughts and suggestions for follow-up work. In section 7 you can find the reference.

1.1 Background

As we all know, the expansion and survival of many companies also strongly depends on the accomplishment of their sales objectives, so sale prediction plays a significant role in business management. For a car dealer, a poor decision about which type of car to invest in may conduct loss of money or even total failure in business. Obviously, the more sale a type of car has, the more profitable this type of car is, and the more reasonable it is to invest in them.

However, since every car dealer wants to invest in more popular cars, it is a very hard job to tell which type of car will have the most sales in advance. Even the most experienced businessman may cannot be able to make perfect decision. The problem lies in two aspects. On one hand, the sales of car are related to too many factors for manual consideration, including but not limited to brand, price, gearbox, structure, emission and current popularity. On the other hand, there are a great number of car models appearing or stopping production every year, making it even harder to predict the sales by experiences.

Since it seems impossible for human to make "perfect" decision, in order to help the car dealers in the task of predicting the future car sales, measures dealing with big data can be applied. It is believed that the law of car sales can be obtained from the data of past sales. In other words, we can build a model to simulate the sales, and find as many as possible features which are believed to have impact on the sales value. With features appropriate enough, the sales predicted can be quite "perfect". In addition, different models usually offer different prediction. It is another challenge to find out the most reasonable result among all the results of different models.

1.2 Problem Definition

The topic title is Auto Car Sales Prediction. In the project, given the car sales data, reviews, etc. over two hundred types of cars for the past ten years, our goal is to build a model to predict the sales volume and revenue in the months of November and December for each type of car.

Specifically, the origin data source includes components as follows. 1) The list of all the car models we will use in the project. 2) Car sales data for ten years (from 2007 to 2016). 3) The Baidu index (which is a value reflecting the dally popularity of a car model) for each type of car over one year (from 2015 to 2016). 4) A set of comments about the cars we are looking into from some online forum over ten years. And we can crawl new data from the internet if necessary.

Our purpose is to build a model to predict the sales volume and revenue for each type of car. The input is type of the car and the time we want. The output is the predicted sales.

To evaluate the prediction, mean squared error will be used as evaluation metric.

1.3 Related Works

There have been a lot of measures for prediction. For example, Zhang G P[1] from Georgia State University introduced a time series forecasting method using a hybrid ARIMA and neural network model. Autoregressive integrated moving average (ARIMA) is one of the popular linear models in time series forecasting. Research activities in forecasting with artificial neural networks (ANNs) suggest that ANNs can be a promising alternative to the traditional linear methods. ARIMA models and ANNs are often compared with mixed conclusions in terms of the superiority in forecasting performance. In the research, a hybrid methodology that combines both ARIMA and ANN models is proposed to take advantage of the unique strength of ARIMA and ANN models in linear and nonlinear modeling. Experimental results with real data sets indicate that the combined model can be an e5ective way to improve forecasting accuracy achieved by either of the models used separately.

Odom M D and Sharda R[2] choose to concentrate on neural network model. One interesting area for the use of neural networks is in event prediction. This study develops a neural network model for prediction of bankruptcy and tests it using financial data from various companies. The same set of data is analyzed using a more traditional method of bankruptcy prediction, multivariate discriminant analysis. A comparison of the predictive abilities of both the neural network and the discriminant analysis method is presented. The results show that neural networks might be applicable to this problem.

Sales prediction is also not a fresh problem. A lot of research has been done on this topic. Delgado-Gómez D, Aguado D, Lopez-Castroman J, et al[3] tried to use support vector machines to improve the sale performance. In their research, an expert system based on support vector machines is developed to predict the sale performance of some insurance company candidates. The system predicts the performance of these candidates based on some scores, which are measurements of cognitive characteristics, personality, selling skills and biodata. An experiment is conducted to compare the accuracy of the proposed system with respect to previously reported systems which use discriminant functions or decision trees. Results show that the proposed system is able to improve the accuracy of a baseline linear discriminant based system by more than 10% and that also exceeds the state of the art systems by almost 5%. The proposed approach can help to reduce considerably the direct and indirect expenses of the companies.

The work of Yu X, Liu Y, Huang X, et al[4] focuses on using review to predict sales. Writing and publishing reviews online has become an increasingly popular way for people to express opinions and sentiments. Analyzing the large volume of online reviews available can produce useful knowledge that are of interest to vendors and other parties. Prior studies in the literature have shown that online reviews have a significant correlation with the sales of products, and therefore mining the reviews could help predict the sales performance of relevant products. However, those studies fail to consider one important factor that may significantly affect the accuracy of the prediction, i.e., the quality of the reviews. In this research, they propose a regression model that explicitly takes into account the quality factor, and discusses how this quality information can be predicted when it is not readily available. Experimental results on a movie review dataset confirm the effectiveness of the proposed model.

Apart from sales of physical goods, Krasonikolakis I, Vrechopoulos A and Pouloudi A[5] focuses on sales in virtual worlds, which have emerged as a new context for gaming, collaboration, social networking but also commercial activity. They explore store selection criteria in virtual world stores and extends earlier research in both offline and online commercial environments, taking into account the novel IT capabilities that VWs exploit. Theoretical insights drawn from the marketing and information systems literature have been used to guide the design of a survey conducted in the virtual world Second Life. In addition to identifying the factors influencing store selection, they investigate how these differ between shoppers and non-shoppers, and identifies the factors that affect the amount of money spent in virtual world shopping environments. The findings suggest that "Core Store Features" and "Security and Privacy" constitute the most important store selection factors in virtual environments and that sales in VWs are predicted by the frequency of visiting and the time spent within VWs' stores.

2. Design

2.1 Overview of Solution

In this section, we are going to discuss about the procedure of our solution to this problem. Here we list out the procedures that we have done, and will have a short discussion in this section. In section 3 and section 4, we are going to talk about the details of our implementation and the problems and the solutions we have faced.

Procedures:

- Observe the data
- Preprocess the data
- Feature engineering
- Model selection
- Model evaluation

In the first step, we are going to have a rough idea of the data we have. Based on that we can decide which kind of data we are going to use and whether the data sets are enough to solve the problem

The second step is to do some pre-processing on the data. After we acquired the data sets, we need to do data integration first which is going to integrate the data sets that are gotten from different sources. Then we need to do data cleaning which is going to deal with the missing data problem and eliminate the outliers. Finally, we need to do data transformation. We need to normalize the data and convert the data into the formats that are suitable for our models.

The third step is to do feature engineering. In this step we need to select the features which are going to be used to train our models. We select features by analyzing the influence of each feature to the result which is the car sales number.

The fourth step is to select and implement our models. In this step we decided to implement three kinds of models which are commonly used to solve the prediction problems.

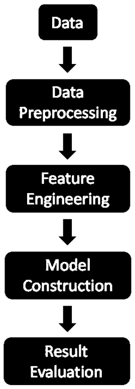


Figure 1 Procedure

The last step is to do the model evaluation. In this step, we can have an evaluation to our models. Based on that, we can enhance our models and have a further discussion about how to generate the final results by using our models.

3. DATA PREPROCESSING & FEATURE ENGINEERING

The first step of our project is to processing the data. Data preprocessing is an important step in data mining and machine learning. Because the data may be out-of-range values, impossible data combinations or missing values, we cannot use the data directly, we have to preprocess the data before analyzing it.

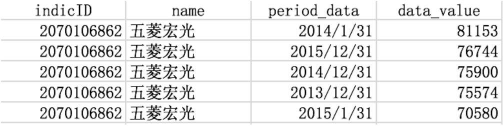
Data preprocessing includes data selection data transformation, data cleaning, data integration, data normalization, etc.

After processing the data, we will do the feature engineering, which plays an important role in constructing the model.

3.1 Data Source

The data source given to us are 3 types of csv files:

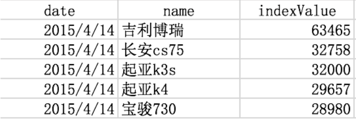
car_sales.csv: sales of 380 types of cars over the past ten years.



indicID	name	period_data	data_value
2070106862	五菱宏光	2014/1/31	81153
2070106862	五菱宏光	2015/12/31	76744
2070106862	五菱宏光	2014/12/31	75900
2070106862	五菱宏光	2013/12/31	75574
2070106862	五菱宏光	2015/1/31	70580

Chart 1 Car Sales Info

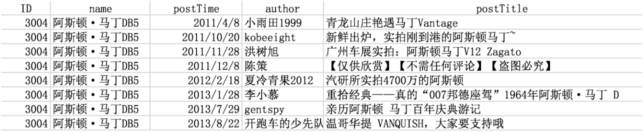
car_bd_index.csv: baidu index of near 400 cars over the past years.



date	name	indexValue
2015/4/14	吉利博瑞	63465
2015/4/14	长安cs75	32758
2015/4/14	起亚k3s	32000
2015/4/14	起亚k4	29657
2015/4/14	宝骏730	28980

Chart 2 Baidu Index Info

car_comment.csv: the comment of near 400 cars on bbs.



ID	name	postTime	author	postTitle
3004	阿斯顿・马丁DB5	2011/4/8	小雨田1999	青龙山庄艳遇马丁Vantage
3004	阿斯顿・马丁DB5	2011/10/20	kobeeight	新鲜出炉，实拍刚到港的阿斯顿马丁~
3004	阿斯顿・马丁DB5	2011/11/28	洪树旭	广州车展实拍：阿斯顿马丁V12 Zagato
3004	阿斯顿・马丁DB5	2011/12/8	陈策	【仅供欣赏】【不需任何评论】【盗图必究】
3004	阿斯顿・马丁DB5	2012/2/18	夏冷青果2012	汽研所实拍4700万的阿斯顿
3004	阿斯顿・马丁DB5	2013/1/28	李小慕	重拾经典——真的"007邦德座驾"1964年阿斯顿・马丁 D
3004	阿斯顿・马丁DB5	2013/7/29	gentspy	亲历阿斯顿 马丁百年庆典游记
3004	阿斯顿・马丁DB5	2013/8/22	开跑车的少先队	温哥华提 VANQUISH，大家要支持哦

Chart 3 Car Comment Info

After observing data, we decided to use the monthly car sales data, monthly average baidu index for each car and monthly average number of bbs comments for each car.

And the data format would be:

[car_name, sales, year, month, time]

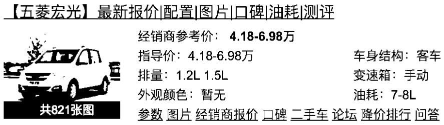
[car_name, year, month, avr_bd_index]

[car_name, avr_comment_num]

3.2 Data Crawling

We find that the data given to us cannot reflect the relationship between different types of cars very well, so we designed to crawl some data for detailed car information from the Internet.

We used python urllib module and regular expression to make a spider to crawl the data from the website *automobilehome*.



【五菱宏光】最新报价|配置|图片|口碑|油耗|测评

经销商参考价： **4.18-6.98万**

指导价：4.18-6.98万　　车身结构：客车

排量：1.2L 1.5L　　变速箱：手动

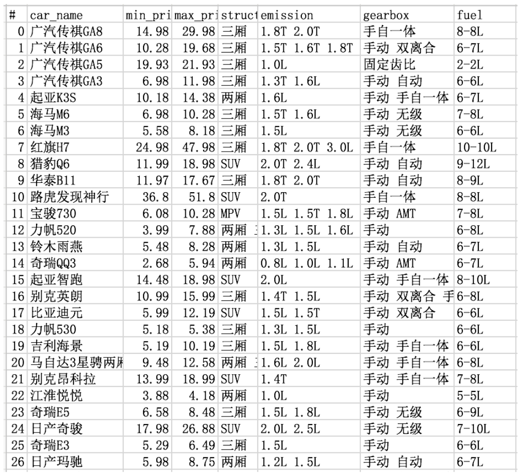
外观颜色：暂无　　油耗：7-8L

参数 图片 经销商报价 口碑 二手车 论坛 降价排行 问答

Figure 2 Car Info

The Data we crawled includes car name, car structure, gear box, emission and fuel consumption.

The data we crawled is like:



#	car_name	min_pri	max_pri	struct	emission	gearbox	fuel
0	广汽传祺GA8	14.98	29.98	三厢	1.8T 2.0T	手自一体	8-8L
1	广汽传祺GA6	10.28	19.68	三厢	1.5T 1.6T 1.8T	手动 双离合	6-7L
2	广汽传祺GA5	19.93	21.93	三厢	1.0L	固定齿比	2-2L
3	广汽传祺GA3	6.98	11.98	三厢	1.3T 1.6L	手动 自动	6-6L
4	起亚K3S	10.18	14.38	两厢	1.6L	手动 手自一体	6-7L
5	海马M6	6.98	10.28	三厢	1.5T 1.6L	手动 无级	7-8L
6	海马M3	5.58	8.18	三厢	1.5L	手动 无级	6-6L
7	红旗H7	24.98	47.98	三厢	1.8T 2.0T 3.0L	手自一体	10-10L
8	猎豹Q6	11.99	18.98	SUV	2.0T 2.4L	手动 自动	9-12L
9	华泰B11	11.97	17.67	三厢	1.8T 2.0T	手动 自动	8-9L
10	路虎发现神行	36.8	51.8	SUV	2.0T	手自一体	8-8L
11	宝骏730	6.08	10.28	MPV	1.5L 1.5T 1.8L	手动 AMT	7-8L
12	力帆520	3.99	7.88	两厢	1.3L 1.5L 1.6L	手动	6-8L
13	铃木雨燕	5.48	8.28	两厢	1.3L 1.5L	手动 自动	6-7L
14	奇瑞QQ3	2.68	5.94	两厢	0.8L 1.0L 1.1L	手动 AMT	6-7L
15	起亚智跑	14.48	18.98	SUV	2.0L	手动 手自一体	8-10L
16	别克英朗	10.99	15.99	三厢	1.4T 1.5L	手动 双离合 手	6-8L
17	比亚迪元	5.99	12.19	SUV	1.5L 1.5T	手动 双离合	6-6L
18	力帆530	5.18	5.38	三厢	1.3L 1.5L	手动	6-6L
19	吉利海景	5.19	10.19	三厢	1.5L 1.8L	手动 手自一体	6-6L
20	马自达3星骋两厢	9.48	12.58	两厢	1.6L 2.0L	手动 手自一体	6-8L
21	别克昂科拉	13.99	18.99	SUV	1.4T	手动 手自一体	7-8L
22	江淮悦悦	3.88	4.18	两厢	1.0L	手动	5-5L
23	奇瑞E5	6.58	8.48	三厢	1.5L 1.8L	手动 无级	6-9L
24	日产奇骏	17.98	26.88	SUV	2.0L 2.5L	手动 无级	7-10L
25	奇瑞E3	5.29	6.49	三厢	1.5L	手动	6-6L
26	日产玛驰	5.98	8.75	两厢	1.2L 1.5L	手动 自动	6-7L

Chart 4 Crawled Data

And the attributes are:

[car_name, price, structure, emmission, gearbox, fuel_consump]

3.3 Data Transformation

3.3.1 Concert string values into num values

We cannot use string values directly to train the model. So we concert string values into num value. For example, for gearbox, we arbitrary let the values be transformed like follows:

Original	Manual	Auto	AMT	Fixed	…
Transformed	1	2	3	4	…

Chart 5 Transformed Value

3.3.2 Convert Attribute with multi values into Multi Attributes with single values

We find that there are many cars has multi values in some attributes. So we convert them into multi attributes with single values. For example, we covert the attribute car_ structure into multi attributes with single values like:

[struct_2x, struct_3x, struct_suv, struct_mpv, struct_bus].

Similar method is used for dealing with attribute car_gearbox and attribute car_emission.

3.3.3 Normalization

Normalization means adjusting values measured on different scales to a notionally common scale. Normalization may reduce the time of processing in machine learning and increase the accuracy of measurement. The most common approach in data normalization is to normalize on the range of whole data set. In our cases which is to normalize according to all the sales volume. However, this may lead to some problems here. Due to the big difference in sales volume between different types of cars, the normalized result of one type of car will change slightly. This indicates that the wave motion won't have a significant change, which may cause the problem that all the predict result of one type of car will be stable no matter what the time is.

Instead of using the origin method to measure on the whole dataset, we choose to normalize on each type of car. Normalize on one type of car have several advantages. The Normalize method using here is defined as:

$$norm\_value = \frac{value - mean}{max - min}$$

It can reflect the wave motion of the cars equally even if the actual sales volume have a big difference. The maximum value in the normalized result will be around 0.5 of all type of cars, which avoid the problem that the car has a large sales volume will have a stronger effect on the result. The result after normalization of all the cars will be mostly in the range of (-0.5, 0.5) which reflect the trend of time more compared to the method of normalize on the whole dataset.

3.4 Data Cleaning

In statistics, an outlier is an observation point that is distant from other observations. An outlier may be due to variability in the measurement or it may indicate error data; the latter are sometimes excluded from the data set. In the data of sales volume, there is low possibility of having variability in the measurement. Hence, the aim of this step is to figure out noise data.

Since the difference of sales volume between different types of autos are considerably large. The better way of outlier detection is to mine under the context of each type of car. We assume it is an outlier if the sale volume is far away from other sales volume and lies alone. Density based method will be used here to figure out the outliers. Like DBscan method, we will define distance and minimum number of points, but measure in 1d space. The distance is defined as:

$$Eps = \frac{Max - Min}{3}$$

Max is the maximum volume of the corresponding car. Min is the minimum sales volume of the car. The minimum number of points appears together is defined as:

$$MinPts = 2$$

3.5 Data Integration

The final step is Data Integration – merging the information together.

At first we planned to join the tables on the attribute 'car_name'. But we find a challenge that there are some cars have different car_names in different tables, while they are the same car. For example, the 'Audi A6' in car_sales table and the 'Audi a6' in car_comments table are exactly the same car.

So we used jaccord similarity to find these cars, and then unified their names.

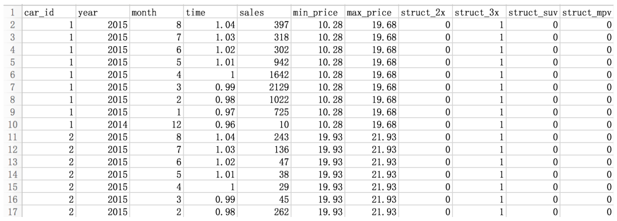
After that, we generate attribute car_id to substitute the attribute car_name.

Since we have already some tables and each table has the 'car_id' attribute. So we join the tables on 'car_id'.

The final attributes are:

[car_id, year,month, time, sales, avr_bdu_index, avr_comment, min_price, max_price,struct_2x, struct_3x, struct_suv, struct_mpv, struct_bus,gb_manual, gb_auto, gb_ma_au, gb_amt, gb_double, gb_fixed, fc_low, c_high, emission_E, emission_0.8L, emission_1.0L, ... , emission_3.6L]

And the final data is like:



1	car_id	year	month	time	sales	min_price	max_price	struct_2x	struct_3x	struct_suv	struct_mpv
2	1	2015	8	1.04	397	10.28	19.68	0	1	0	0
3	1	2015	7	1.03	318	10.28	19.68	0	1	0	0
4	1	2015	6	1.02	302	10.28	19.68	0	1	0	0
5	1	2015	5	1.01	942	10.28	19.68	0	1	0	0
6	1	2015	4	1	1642	10.28	19.68	0	1	0	0
7	1	2015	3	0.99	2129	10.28	19.68	0	1	0	0
8	1	2015	2	0.98	1022	10.28	19.68	0	1	0	0
9	1	2015	1	0.97	725	10.28	19.68	0	1	0	0
10	1	2014	12	0.96	10	10.28	19.68	0	1	0	0
11	2	2015	8	1.04	243	19.93	21.93	0	1	0	0
12	2	2015	7	1.03	136	19.93	21.93	0	1	0	0
13	2	2015	6	1.02	47	19.93	21.93	0	1	0	0
14	2	2015	5	1.01	38	19.93	21.93	0	1	0	0
15	2	2015	4	1	29	19.93	21.93	0	1	0	0
16	2	2015	3	0.99	45	19.93	21.93	0	1	0	0
17	2	2015	2	0.98	262	19.93	21.93	0	1	0	0

Chart 6 Processed Data

3.6 Splitting Training Data & Test Data

We split the training data and test data by time point Sep, 2015. We use the data before Sep, 2015 as the training data to train the model, and use the recent 1-year (from Sep, 2015 to Aug, 2016) data as the test data to evaluate the model.

And we plan to use all the data as training data to train the model for predicting sales volume in the future.

3.7 Feature Engineering

In this part, we are going to discuss about the feature engineering. As we all know, feature engineering is an important step for the model constructing. We try to find out the features that are corresponding to the final results.

First, we are interested in the distribution of the sales value based on the specific feature values.

Intuitively, time related features should be the first we should consider. Firstly, we draw a diagram based on the average sales values of all cars in each month and each year (from 2007 to 2016).

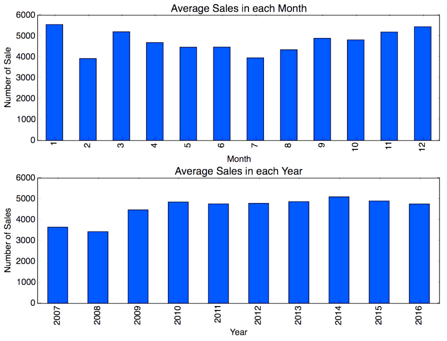


Figure 3 Year and Month Feature Analysis

In this diagram, we can find out that the sales values are strongly influenced by the time. For example, the average sales value of February is the lowest of the whole year also we can see that the sales values are decreasing from March to July, but they are raising from July to December. This means the month feature is important for each kind of models. Meanwhile, we can see that the sales values are influenced by the year. During the first two years, the values are lower than other years, and the average values are almost the same in the last few years. This tells us, based on the average sales values of the last year, we can give out a relatively precise result.

Here we have another example to illustrate how do we use the comments information. At the beginning, we are trying to find out the actual meaning of the comments information. But after we observed some comments of some well-sale cars and some cars with the low sales value, we find that the comments often do not have some useful meaning. But the number of comments usually can reflect the population of this kind of car. Which means, we may use the number of comments of each car as the feature to train our models.

But the question is what is the relationship between number of comments and the sales values. Based on this question, we draw a diagram below.

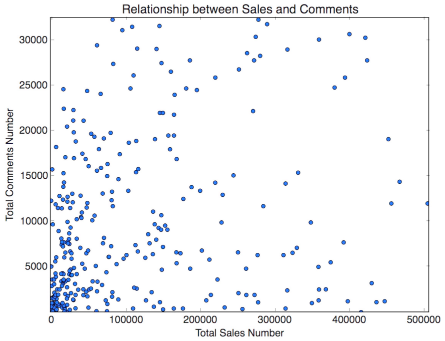


Figure 4 Comment Number

This diagram shows that there is no obvious relationship between the comments number and the sales values. But we can see that some of the cars that have high sales values may not have many comments. The same things happened to some cars with low sales values but have many comments. This tells us this feature may not have the simple relationship between the final results. It may require more information to get the influence of this feature.

Based on the same kind of method, we can evaluate each feature that we have now. These features are divided into two kinds of features. The features that have obvious influence on the results and the features that may not have influence on the results.

Based on these two kinds of features, we will do some experiments to train our models. For example, we may first input those features that have obvious influence on the results into our models. And then, try to input some of the other features to test weather it will give a better result.

4. Models

When choosing models, we have to consider the diversity of models. So we chose three different types of models – Linear Regression, Naïve Bayes and Multilayer Perceptron Regression. Linear Regression is a linear model. Naïve Bayes is a model based on probability. And the Multilayer Perceptron Regression is a non-linear neural network model.

4.1 Linear Regression

The first model we chose is the linear regression model which is quite simple to understand and widely used in prediction problems. From its name, we can easily tell that it is a kind of linear model. The idea of this model can be defined by the formula below. This model has some assumptions [1] which also determine the result of this model. There are three important assumptions:

4.1.1 Linearity

This gives the assumption on the mean value of the response variable. It is treated by a linear combination of the parameters and the predictor variables.

4.1.2 Constant Variance

This assumption is about the errors. It assumes different variables have the same variance in their errors.

4.1.3 Independence of Errors

This means the errors of different response variables are independent with each other.

Based on these assumptions, the first thing we need to do is to select the features that we are going to train the linear regression model. At the beginning, we are trying to use all the features of the call information. We assume that the information of a car is independent with each other and the influence of them is a linear combination.

Here we are facing one problem, it is how to deal with time related information such as monthly sales number, monthly Baidu index and monthly comment number.

The date information is in the format of year and month. If we directly input year and month, we may lose some information because year and month should not be independent from each other. For example, the December of 2015 should have some relation with January of 2016. To solve this problem, we create a new feature which is called time by computing year * 12 + month.

After that we perform normalization on these features to make sure they are in the same distribution range. Also we need to do the normalization as we discussed in section 3 on the output result.

For the implementation, we use python with scikit-learn package and pandas as the data processor. By giving the input features and output results, we can train a linear regression model and use this model to predict the test data. After that we can get the prediction results. In the evaluation section, we are going to use these results to evaluate and modify our models.

4.2 Naive Bayes

In machine learning method, Naive Bayes is a simple probabilistic classifier based on applying Bayes theorem with strong independence assumptions between the features. It is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. Naive Bayes was first introduced under the text retrieval domain, and remains a baseline method for text categorization, the problem of judging documents as belonging to one category or the other with word frequencies as the features. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations.

In our case, we will use Gaussian Naive Bayes model, for some of the attributes are continuous data and the distribution conform to Gaussian Distribution.

Naive Bayes is a conditional probability model: given a problem instance to be classified, represented by the vector $X = \{x_1, x_2, \ldots x_n\}$ where x_n represents the n^{th} independent feature. The probability of the result is represented as $P(C_k|X)$ where C_k denotes the k^{th} classifier. According to Bayes Theorem, $P(C_k|X)$ can be represented as:

$$P(C_k|X) = \frac{P(C_k)\, P(X|C_k)}{P(X)}$$

Here, in our model, X denotes the features we get from the data. The sales volume is assigned to 10 buckets according to the normalization result. $P(C_k|X)$ denotes that given the features X, the possibility of the price is in the k^{th} bucket.

According to the normalization we use, the way of bucketing is an equal width bin method. The reason why we didn't select equal depth bin is because if we use equal depth bin, the bin with high value will have a large range of representation. For the reason that Naive Bayes can only predict the bucket not the specific value, this may cause a problem that if the bucket has a large range, the difference between the max and the min value is large, which may lead to the decrease of accuracy.

Different from other models, it is not beneficial to use the features that are sparse which may cause a bias on the result that has more records. According to all the features we select, many of them, due to separation from multi-value, are sparse. Naive Model is better not to select these features, or use directly from the multi-value features.

4.3 Multilayer Perceptron Regression

The last method we choose is Multilayer Perceptron (MLP). It is a feed-forward artificial neural network model that maps sets of input data onto a set of appropriate outputs. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes the supervised learning technique back-propagation for training the network.

Multi-layer Perceptron Regression is a regression implementation. It trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters. It can also have a regularization term added to the loss function that shrinks model parameters to prevent over-fitting.

For the implementation, we use a python module sklearn. neural_network.MLPRegressor from the machine library scikit-learn to train our model. This module works with data represented as dense and sparse numpy arrays of floating point values.

We use module pandas.dataframe to process the data. We let the data in the training_data.csv dropping sales data as the X_train data and the sales data as y_train data. Similarly, we let the data in the test_data.csv dropping sales data as X_test data and the sales data as y_test data.

We enhanced the MLPR's result by modifying the size of hidden layers and changing the numbers of the iteration.

5. Model Evaluation

5.1 Metrics

5.1.1 Mean Square Error

Mean squared error measures the average of the squares of the errors or deviations which is the difference between the estimator and what is estimated. Mean squared error is a risk function, corresponding to the expected value of the squared error loss. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

In this situation, we use 1 year of data as test data. The error is the difference between the sales volume and the predict from our result. Because the sales volume may have a range of 100000, the Mean Square Error may have a relatively big result on some types of the cars. It can't reflect the result directly, so we use another method to evaluate the result.

5.1.2 Histogram

According to the normalization method we use, the car sales volumes are mostly mapped to the range of (-0.5, 0.5) unless the data is skewed. Every 0.1 in the normalized result denotes 10% of the difference between the max and min volume of each type of car. In order to have a direct view of our result, we need to map them in several buckets and accumulate the results. The method here is:

$$B_i = Round(\ Diff{*}10)$$

Here, B_i is the bucket that current result will be assigned to. $Diff$ is the difference between the prediction and the normalized car sales volumes. $Round()$ function means to round the decimal to the nearest number. We use $P(-2 \leq B \leq 2)$ to measure as accuracy, which means the distance of predicting result and sales volume lies within 2 buckets.

5.2 Result Evaluation

In this section, we are going to do the evaluation after we train our three models. As we discussed in the previous section, we will use the data of last year (2016) as the test data set and the rest of data as the training data. So our goal is to predict the sales values of each car in 2016. The results of this test can be used to evaluate our models and do some further adjustment.

5.2.1 Linear Regression

First we draw a graph based on the test results of the linear regression model. The Y-axis is the number of samples which are the sales values of one car in one month. As for the X-axis, it is the error rate based on the difference between maximum sales value and the minimum sales value of the car itself. For example, if one sample falls into the bin 1.0, it means that the predication value of one car on one month is higher than the difference between the maximum and minimum sales value of this car (based on the entire data set) about 10 percent.

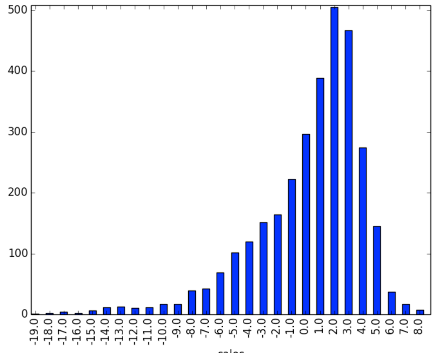


Figure 5 Linear Regression Result

After we understand the meaning of this graph, we can tell that the accuracy of the linear regression model is not good enough. About 49.312% of samples are in the range of -2 to 2. Also we can see that the most of samples have the error rate which is greater than 0. This means the linear regression model tends to predict the results higher than the actual values.

In addition, we can use the common mean square error to measure the model which means we take the average value of the square of the difference between predict value and actual value. The mean square error of the linear regression model is 46787444.3361. This result shows that the performance of linear regression model is not good enough for our assumption. The influence of the features to the results may not be the linear combination.

Based on this result, we can use the linear regression model as the baseline of our prediction. Further more, we may not use the results of linear regression model as a part of our final output. Instead, we can use the results as the baseline.

5.2.2 Naive Bayes

The following figure shows the results of using all the features we have.

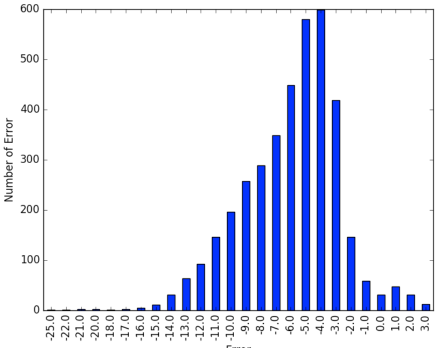


Figure 6 NB Result for All Attributes

Significantly, it is not a good result. What we assume to be the peak of the histogram should be 0. However, the peak in this histogram is around -4, which means the predicting result and the sales data has a difference of $0.4\ *\ (max - min)$ according to the car. It is 40% of the maximum volume minus the minimum volume of the type of the car which is believed to be unacceptable by us. The threshold we believe can be acceptable is 2. Thus, the accuracy which is $P(-2 \leq diff \leq 2)$ is 0.417. The Mean Square error is 54593287.

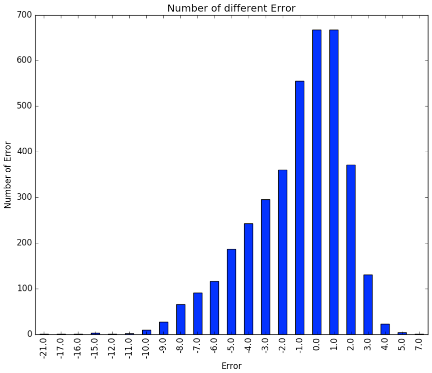


Figure 7 NB Result for Best Result

To optimize the model, we need to adjust the features we use. Though all the features in the data set is believed to be relative, some of the features may not have a positive effect to the result.

We keep the positive features which have a positive effect on the accuracy and mean square error and eliminate negative features which may decrease the accuracy and increase mean square error. The following figure shows the best result we can get using Naive Bayes model. The best features selected in the model is $\{month, year, structure, price\}$. The accuracy is 0.685 and the mean square error is 23475463.

5.2.3 Multi-layer Perceptron Regression

The following figure shows the results of multilayer perceptron regression.

In multilayer perceptron regression model, about 68.9% of samples are in the range of -2 to 2. Most of samples have the error rate which is greater than 0, which indicates our multilayer perceptron regression model tends to predict the results higher than the actual values.

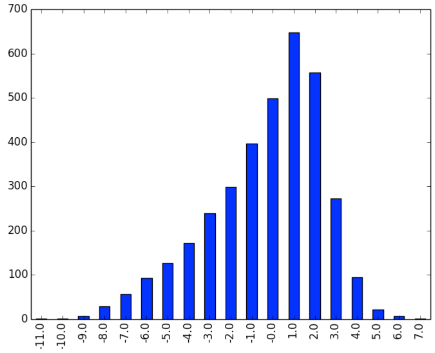


Figure 8 MLPR Result

We also use the common mean square error to measure the model. The mean square error of the multilayer perceptron regression model is 15325963, which is the smallest among the three models we choose. It is obvious that the multilayer perceptron regression model has the best performance.

6. CONCLUSION

6.1 Summary

In this project, we learnt some techniques in big data and we have built a model to predict the sales volume and revenue for each type of car given car sales data and reviews data. First we observed the data given and crawled some data we need from the Internet. Then we preprocess the data we have, including data transformation, data cleaning, data integration and data selection, etc. After processing the data, we extract features from the data. And then we choose three different types of models – Linear Regression, Naïve Bayes and Multilayer Perceptron Regression. Linear Regression is a linear model. Naïve Bayes is a model based on probability. And the Multilayer Perceptron Regression is a neural network model. We predict the sales volume by each model and showed the results of each one.

6.2 Lessons Learned

In the procedure of this project, we obtained a lot of experience.

For example, we learned to make sure that all the features in our model do make sense. At the very beginning, we thought that the more feature we extract, the more accurate the model will be. So we used all the data provided by the project, which is introduced in section 1.2, for we thought all they have something to do with the sales. What's more, we crawled some data from online sources, which including the review from some forums about the cars. However, after the models are built, we found that the comments and reviews are not good features. Because a great proportion of the comments and reviews are redundant or unrelated information (for example, chat among forum users). Adding the comments and reviews as features will make the model less accurate. After that, we understood that the features should be proved reasonable enough to be added to the model.

In addition, through this project, we understood the procedure to do prediction using regression analysis. We acquired more knowledge about several regression models, including linear regression, naïve Bayes model and multilayer perceptron regression.

6.3 Future Work

From the result analysis, we find that Linear model and MLPR model tends to predict higher in result and that Bayes mode tends to predict lower in result. So we assume that there would be a possible way to improve the accuracy using a hybrid model. Thus, we plan to use some algorithms like Adaboost to add weight to each model to construct a new hybrid model.

7. REFERENCES

[1] Zhang G P. Time series forecasting using a hybrid ARIMA and neural network model [J]. Neurocomputing, 2003, 50: 159-175.

[2] Odom M D, Sharda R. A neural network model for bankruptcy prediction[C]//Neural Networks, 1990., 1990 IJCNN International Joint Conference on. IEEE, 1990: 163-168.

[3] Delgado-Gómez D, Aguado D, Lopez-Castroman J, et al. Improving sale performance prediction using support vector machines [J]. Expert systems with applications, 2011, 38(5): 5129-5132.

[4] Yu X, Liu Y, Huang X, et al. A quality-aware model for sales prediction using reviews[C]//Proceedings of the 19th international conference on World Wide Web. ACM, 2010: 1217-1218.

[5] Krasonikolakis I, Vrechopoulos A, Pouloudi A. Store selection criteria and sales prediction in virtual worlds [J]. Information & Management, 2014, 51(6): 641-652.

[6] Shaw, 1992 J. Shaw Neural network resource guide AI Expert, 8 (2) (1992), pp. 48–54.

[7] Tsay R S. Analysis of financial time series[M]. John Wiley & Sons, 2005.

[8] Enke, D., & Thawornwong, S. (2005). The use of data mining and neural networks. Expert Systems with Applications 29, 927-940.

[9] Liang, J., Song, W., & Wang, M. (2011). Stock Price Prediction Based on Procedural Neural Networks. Advances in Artificial Neural Systems.

[10] Bahia Itedal, S. H. (2013). A Data Mining Model by Using ANN for Predicting Real Estate Market: Comparative Study. International Journal of Intelligence Science 3, 162-169.