Establishing a Big Data Analysis Framework And Computing Equilibrium Market Share with Vehicle Data

Si-Jheng Ciou, Yi-Hau Huang, and Yu-Ching Lee†
Department of Industrial Engineering and Engineering Management
National Tsing Hua University, Hsinchu, Taiwan
Email: zxc940907@gmail.com, borrorboy@gmail.com, yclee@ie.nthu.edu.tw

Abstract. Data analysis on a scalable framework has already been a popular and widely employed technique for many years. The technique allows the processing and analysis of data that is unable to be dealt with by the general software. Hadoop is one of the robust and well-behaved tools in views of parallel computing and distributed storage. We employ big data analysis to discover the potential information and patterns on Hadoop. The aim is to generate a quantifiable value to represent the customers' willingness to buy products. Our case study is the vehicle data. First, we split the whole data into training data and testing data. Second, we combine Hadoop with R so that we can analyze the training data on R by keying in the commands of Hadoop Distributed File System. Third, the equilibrium of customers' choice on R platform is solved. Finally, the proposal method is further validated by the real data of vehicles. This project is aimed to provide a method to connect the advanced game-based model with the existing scalable computing system.

Keywords: Big Data Methodology and Applications

1. INTRODUCTION

The study surrounding big data has become very popular during this decade. The development of *scalable* storage and access technologies allows the analysts to build their own data-handling systems from data collection, storage, extraction, accessing, managing, statistical operation, to advanced computations on clusters of computers. The main techniques used for storage and access are the DFS and the MapReduce. The DFS is the distributed file system and the MapReduce is a distributed computing structure named from two critical functions Map and Reduce in the procedure.

The Hadoop framework remains the most popular open-sourced framework which employs the Hadoop Distributed File System (HDFS) and the Hadoop MapReduce. On the Hadoop framework, there is flexibility to install and set up a user-defined pipeline for the data analytics depending on the types of data and the end use. The collection of the software and the components is often referred to as the *Hadoop ecosystem*. The ecosystem (Sitto and Presser, 2015) is comprised of core technologies, database, data management, serialization, monitoring, analytics and machine learning libraries, data transfer, security, access control, auditing, cloud computing and virtualization. The pipeline of data analysis is often represented by technologies stack.

Spark is another popular new-generation open-sourced big data framework. It significantly improves the original Hadoop MapReduce procedure by in-memory cluster computing capabilities and thus reduces the hard disk Input/Output. This improvement is beneficial to the iterative computation particularly those requiring information carrying. Because many resources have been invested in the HDFS-format data, Spark was developed in a way to be compatible with (rather than replace) the Hadoop ecosystem. Spark allows access to any data format of the Hadoop ecosystem, e.g. HDFS, YARN, Hive, HBase, etc. Even though the project is young, Spark is currently widely-believed (Karau et al., 2015) to be a suitable framework for developing scalable computational algorithms.

The value of big data is extracted by the analytics methods. These analytics methods are generally three: statistical inference, data mining, and machine learning. The differences of these methods are minor and mostly based on assumptions and viewpoints. Methods of statistical inference can further be divided into parametric, non-parametric, and semi-parametric analysis. Data mining is a collection of all means and tasks for performing *knowledge discovery in databases (KDD)*. In the framework of machine learning, methods can be classified into supervised learning, unsupervised learning, and reinforcement learning. Although three fields overlap considerably in terms of adopting similar pools of algebraic methods, probability theory, statistics techniques, and optimization, each field grows and develops with their own societies and publications. The representative components of analytics in the scalable framework are RHadoop, Mahout, Spark MLlib, and SparkR. They are the applications for analyzing the data in the HDFS-format.

Since the interpretations of data are not limited to the above structures or utilization, the value of big data that has been explored is just the tip of an iceberg. In this paper we study one of the potentially valuable information of big data, that is, the economical phenomenon *Nash equilibrium*, and build the framework from basic data input to the scalable computation. Since Nash defined and formulated the outcome of consumer behavior and market response as the equilibrium of the games, equilibrium has been the most important notion for analyzing the optimal strategies and predicting other players' actions in a competition.

The Nash equilibrium by definition is a set of strategies optimal to all the participants in the game. The mathematical expression of the Nash equilibrium is often specified as a satisfactory solution to a *variational inequality (VI)* or a *complementarity problem (CP)*. Fitting the recorded observations, i.e., actions taken in reality, by the VI or CP models according to the *least squares of error* criterion is equivalent to solving a nonconvex nonlinear programming optimization problem. Solving for global optimum of such a program is among the most difficult problems in this age, but solving for local optimum or a stationary point is a tractable task with standard approaches such as quasi-Newton method, conjugate gradient (CG) method, and sequential quadratic programming (SQP) method.

Our specific aims in this project are as follows:

1. Expand the analytical capability of the Hadoop ecosystem to the computation of an economical invisible phenomenon.
2. Establish the big-data-conceptual formulation for computing Nash equilibrium of a given market, here the vehicle market in UK.
3. Develop the MapReduce scalable algorithm to solve the equilibrium estimation problem.

2. RELATED WORK

Compared to a decade ago, the mindset of big data has been gradually firmed up nowadays. Before people have enough preparations for harnessing big data, the major characteristics of big data such as high volume, velocity, and variety, were considered unrealistic to be handled (McAfee and Brynjolfsson, 2012.) Nevertheless, it's not very difficult for people now to recognize this possibility. The hardware and software capability has dramatically progressed, and this progress has made the vision of big data more concrete.

Big data analysis could be applied in the fields of different category, such as e-commerce & market intelligence, e-government & policies, science & technology, smart health & wellbeing, and security & public safety (Chen et al., 2012.) Focusing on the e-commerce and market, Amazon and eBay, which are the leaders of e-commerce vendors, are using big data in highly scalable e-commerce platforms and product recommender systems by the techniques such as text mining and graph mining (Pang and Lee, 2008; Adomavicius and Tuzhilin, 2005). The major Internet companies, like Google, Amazon, and Facebook, devoted a large number of resources to the web analytics, cloud computing, and social media platforms. Firms also can reach the niche markets because of sufficiency of information generated by the analysis of big data (Brynjolfsson et al., 2006.)

There are many commercial Hadoop platforms that have been built in recent years, such as IBM BigInsights, Cloudera, Amazon Web Services (AWS), Microsoft HDInsight, Intel Distribution for Apache Hadoop, etc.

Economical phenomenon by nature is an unseen pattern of the population. Transactions of the commodity consumption form the streaming data of very large scale. Nash equilibrium is an unseen quantity, and in the past, the measurement of the Nash equilibrium can only be computed based on a macro view with less observations. A survey of the computation of equilibria can be found in McKelvey and McLennan (1996). The potential of explaining economic issues with the utilization of big data is noticed by Einav and Levin (2014), where the rise of empirical economics is addressed and its connection, especially the minor distinction, with the machine learning approach is argued.

The method of structural estimation (Su and Judd 2012) that estimates the unknown parameter under an assumed population and market equilibrium by a constrained optimization model has made a great impact and caused a great deal of debate (Iskhakov et al. 2016) in the economics society. Following the similar line of structural estimation and the same assumption on consumers' utility (which is assumed to be a pure characteristics demand model (PCM)), our work is a scalable extension and a real-data implementation of the endogenous price formulation in Pang et al. (2015) on Hadoop.

3. ESTABLISHING THE BIG DATA ANALYSIS FRAMEWORK FOR VEHICLE DATA

Despite the existence of commercial Hadoop framework on cloud, for this project we aim to establish scalable data analysis framework with the open-sourced components on the local machines and self-defined data processing pipeline. The purpose is to comprehend the Hadoop infrastructure and assist us to realize the academic application. We are going to discuss the pipeline established in our laboratory in Section 3.1. The Sections 3.2 is the description of data and data cleaning, respectively. Eventually, we will describe the preliminary data-mining for the vehicle data in Section 3.3.

3.1 Pipeline

The components in the pipeline we have established include virtual machine, virtual bridged network, HDFS, Hadoop MapReduce, and R.

Virtual Machine

A virtual machine is a software that enables multiple operating systems to run on one powerful computer as if they are on multiple separate computers. The technique of virtual machine is fully developed, and there are many choices of virtual machines like VMWare, Xen, and VirtualBox. VMWare is a paid software. Open sourced Xen does not provide graphical user interface (GUI). Free VirtualBox with GUI is easily and quickly picked up, and it is suitable for the experimental purpose we focus on. We choose the Oracle VM VirtualBox to build our cluster with four virtual machines on two computers. One virtual machine is set as the *master node*, and the others are the *slave nodes*. The virtual machine economically utilizes the resources of the physical machine, and by partitioning the resources of the physical machine into virtual machines we can mimic the behavior of a cluster of physical machines at the cost of only one computer.

Virtual Bridged Network

The bridged network grants the IP address to the virtual machines so they have the ability to communicate with the physical network. Namely, the computers that set up the bridged network is able to communicate with another computer by the IP of a bridge. In the environment of our cluster, all of virtual machines use the bridge interface to link and communicate with each other through a core switch. The master node thus has the channel to communicate with those slave nodes. As a result, distributed computing can be executed on Hadoop framework on each node. Intuitively, the network layout will greatly influence the transmission of a packet, and different network interfaces are required according to the different network layout.

HDFS

Hadoop Distributed File System (HDFS) being the foundation of the Hadoop technology stack is responsible for the distributed storage of files on multiple nodes. HDFS is defined on the system comprised of the *NameNode* (always set on the master node) and the *DataNodes* (set on either the master node or the slave nodes.) The NameNode is used to modify and acquire the definition of files whereas the DataNodes are employed to the physical input/output operations. Based on the HDFS, the Hadoop framework is able to allocate the fragments of the divided files to be stored in each DataNode. With replicas of the divided files, the access of the data is of high reliability (cf. Shvachko, 2010.) HDFS is the basis on which the MapReduce function can be performed. HDFS supplies the important features—fault tolerant, highly reliable, scalable, and simple for expansion—demanded by the MapReduce function. HDFS is well suited for scaling out and is critical to realize big data analysis.

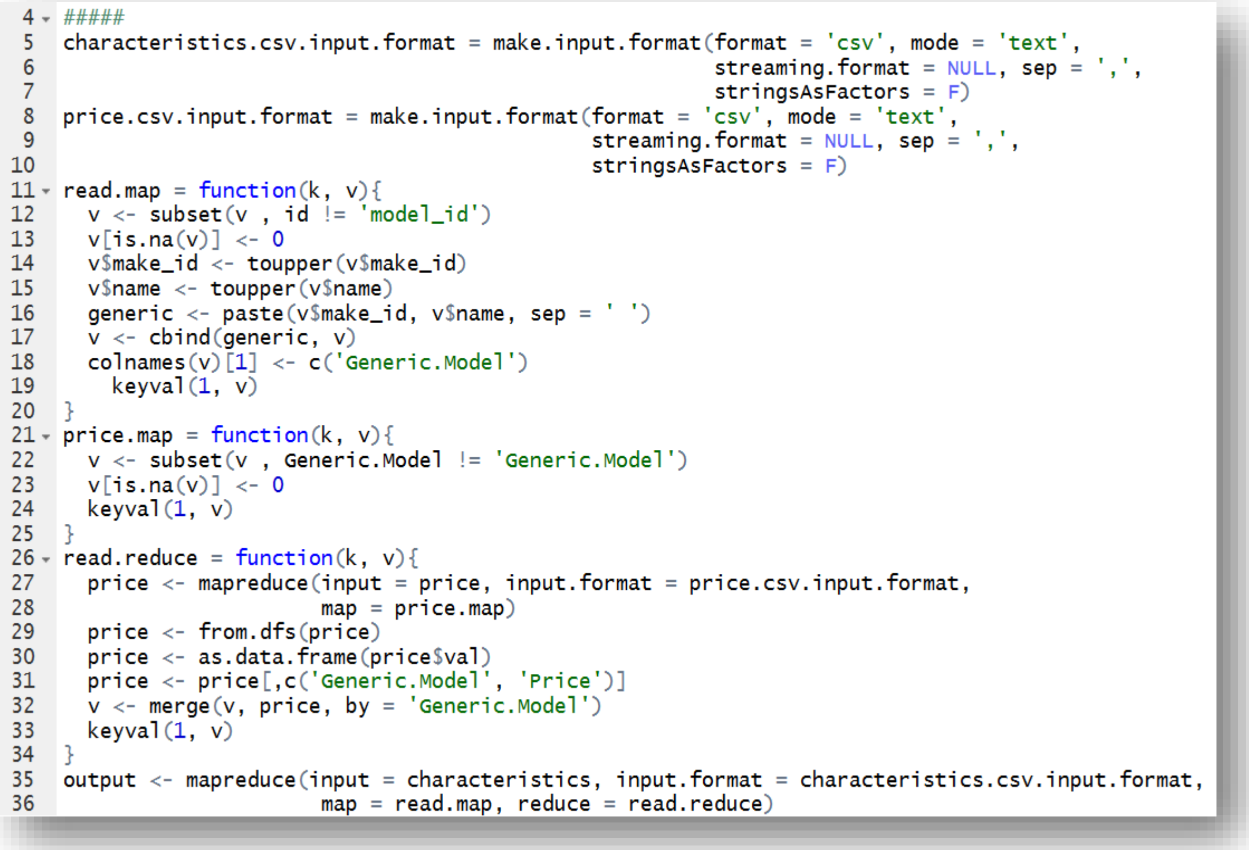
Hadoop MapReduce and R

The MapReduce technique (developed firstly by Google and then by Yahoo) is formed with the Map function and the Reduce function. Briefly speaking, the Map function is to map the divided fragments to the DataNodes for distributed computing. The Reduce function is to reduce the result of Map function and output an outcome (cf. Dean and Ghemawat, 2004.) Concluding the descriptions thus far, the Hadoop framework combines the distributed storage and the distributed processing which are HDFS and Hadoop MapReduce, respectively, to realize the ability for analyzing the enormous data.

With Hadoop MapReduce, we are able to go through the procedures of data mining and mathematical programming that we can't do with the general software restricted by the large size data. Figure 1 displays a partial script of Hadoop MapReduce for data cleaning on the RStudio. As we can see in the script, Hadoop MapReduce allows us to read data of large-scale using the map function. The Hadoop MapReduce will read those divided files as many blocks, process the code on blocks, and finally, return the result to the master node. That is to say, we can filter data to information until we are able to read the data of reduced-size with general software, or we can directly use the huge data to compute the solution to the mathematical programming.

3.2 Data Collection and Cleaning

We use the open statistics of cars registered for the first time by its generic model from the website of Department of Transport, UK. The raw data are collected annually and its time scale is 2001 through 2015-Sep-30. The quantities of cars registered for the first time will later act as the vehicle sales in the market equilibrium computation. The dimension of this data set is of 2,386 rows (generic models) and 16 columns (years). The other data set we use is the characteristics of all models of vehicles of different brands. The vehicle characteristics for different models produced 1936 through 2016 by all motor companies are collected in this data set. The dimension of the data set is of 63,185 rows (models) and 37 columns (characteristics). The self-explanatory labels for every column are model_id, model_make_id, model_name, model_trim, model_year, model_body, model_engine_position, model_engine_cc, model_engine_cyl, model_engine_type, model_engine_valves_per_cyl, model_engine_power_ps, model_engine_power_rpm, model_engine_torque_nm, model_engine_torque_rpm, model_engine_bore_mm, model_engine_stroke_mm, model_engine_compression, model_engine_fuel, model_top_speed_kph, model_0_to_100_kph, model_drive, model_transmission_type, model_seats, model_doors, model_weight_kg, model_length_mm, model_width_mm,



```r
#####
characteristics.csv.input.format = make.input.format(format = 'csv', mode = 'text',
                                                     streaming.format = NULL, sep = ',',
                                                     stringsAsFactors = F)
price.csv.input.format = make.input.format(format = 'csv', mode = 'text',
                                           streaming.format = NULL, sep = ',',
                                           stringsAsFactors = F)
read.map = function(k, v){
  v <- subset(v , id != 'model_id')
  v[is.na(v)] <- 0
  v$make_id <- toupper(v$make_id)
  v$name <- toupper(v$name)
  generic <- paste(v$make_id, v$name, sep = ' ')
  v <- cbind(generic, v)
  colnames(v)[1] <- c('Generic.Model')
    keyval(1, v)
}
price.map = function(k, v){
  v <- subset(v , Generic.Model != 'Generic.Model')
  v[is.na(v)] <- 0
  keyval(1, v)
}
read.reduce = function(k, v){
  price <- mapreduce(input = price, input.format = price.csv.input.format,
                     map = price.map)
  price <- from.dfs(price)
  price <- as.data.frame(price$val)
  price <- price[,c('Generic.Model', 'Price')]
  v <- merge(v, price, by = 'Generic.Model')
  keyval(1, v)
}
output <- mapreduce(input = characteristics, input.format = characteristics.csv.input.format,
                    map = read.map, reduce = read.reduce)
```

Figure 1: Part of the MapReduce procedure in R.

model_height_mm, model_wheelbase_mm, model_lkm_hwy, model_lkm_mixed, model_lkm_city, model_fuel_cap_l, model_sold_in_us, model_co2, and model_make_display.

The datasets of the car registrations by model and of the vehicle characteristics both contain missing and inconsistent data. Conducting data cleaning for improving the quality and integrity of data is essential to assure the effectiveness of the models that are later built on the base this data. Missing data and unknown data can be deleted or be compensated by multiple ways. The empty values of the missing and unknown data in our data sets are compensated with zero. In addition, the total sales from 2011 to 2015-Sep-30 that are less than 25 are deleted. These small numbers of total sales suggest that the corresponding vehicle models are the limited editions or the commemorative editions which are not actually reflecting the real driving habit and transportation demand. After cleaning, the sum of the sales in accordance with different brands are calculated and visualized in Figure 2.

The dataset of vehicle characteristics contains both qualitative data (of categorical scale) and quantitative data (of ordinal, interval, and ratio scale.) For instance, the model_engine_fuel column contains the entries of gasoline, hydrogen, electric, and so on. The empty values occur in the raw dataset have been thoroughly compensated by looking up information from other sources. After compensating, the qualitative columns are transformed into quantitate columns since the input needed for equilibrium computation is completely numerical. Next, data is normalized to take the values between 0 and 1 for every column. Normalization prevents the huge differences in the order of magnitude when comparing values to each other. Finally, the dataset of vehicle sales and the dataset of characteristic for different brands with multiple models are combined, and two extra columns vehicle price and market share added. After this preparation, the dimension for the data is of 212 rows (models of popular vehicles) and 23 columns (vehicle price, market share, and 21 characteristics.) Part of the data is shown in Figure 3.

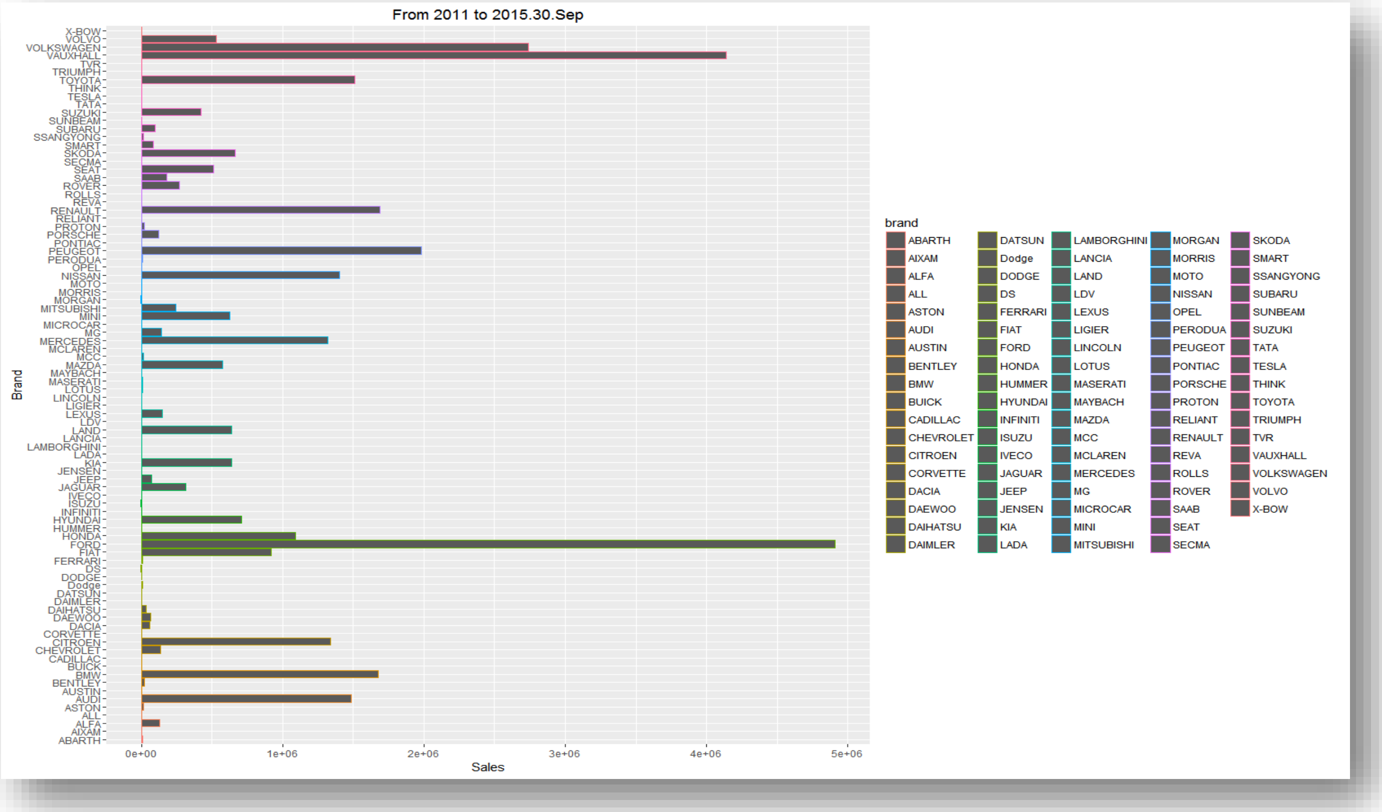


Figure 2: Sales for different brands

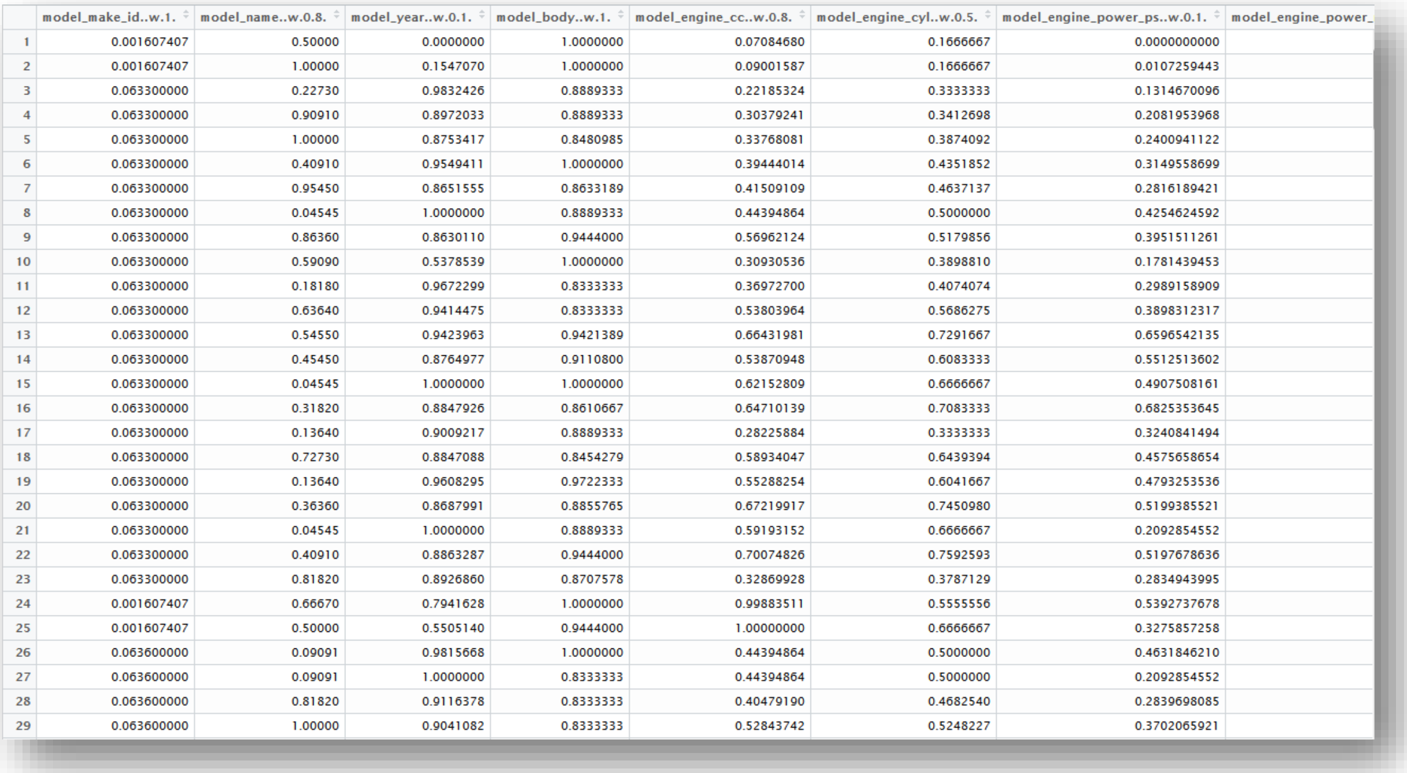
3.3 Data Mining

Data mining can assist the extraction of meaningful patterns and rules in the dataset, and it includes many kinds of analysis techniques and applications. Clustering analysis identifies several clusters based on similarity and dissimilarity. One of the Clustering analyses is k medoids. Compared to other cluster analyses, such as k means, k medoids is not easily impacted by noise and outliers. The result for k medoids is shown in Figure 4. The plot suggests that the vehicle sales data is the best to be divided into 3 clusters. The first cluster is the lowest amounts of sales among the three clusters. The third cluster is the highest amount of sales, but the numbers of vehicle models belonging to the third cluster is the least, which implies that the numbers of the biggest sellers are few.

Time series analysis aims to analyze the features of historical data collected as a time series to discover the trends and provide insight to forecast the future values. For example, under the car brand SEAT there are several models: ALHAMBRA, ALTEA, AROSA, CORDOBA, EXEO, IBIZA, LEON, MII, and TOLEDO. The method establishes a plot of annual sales to observe the trend visualized in Figure 5.

Principal component analysis (PCA) is a kind of data-reduction technique, its purpose is to transform a larger number of correlated variables into a smaller set of uncorrelated variables. We take the columns of model_year, model_engine_position, model_engine_type, model_engine_valves_per_cyl, model_engine_power_ps, model_engine_torque_nm, model_engine_bore_mm, model_engine_stroke_mm, model_transmission_type, model_lkm_hwy, model_lkm_mixed, model_lkm_city, model_fuel_cap_l, model_sold_in_us to establish the principal component analysis as depicted in Figure 6. The plot suggests that the numbers of principal components (with eigenvalues more than 1) should be four. The generated principal components are listed in Figure 7. It can be seen that the principal component 1 (RC1) is highly related with model_engine_power_ps (0.21), model_engine_toeque_nm (0.27), model_engine_bore_nm (0.30), model_engine_stroke_nm (0.20), model_fuel_cap_1 (0.28), and model_sold_in_us (0.13).

Factorial analysis (FA) focuses on discovering the potential structure and identifying the latent or unobserved factors. We take the same columns as for PCA to conduct the



	model_make_id..w.1.	model_name..w.0.8.	model_year..w.0.1.	model_body..w.1.	model_engine_cc..w.0.8.	model_engine_cyl..w.0.5.	model_engine_power_ps..w.0.1.	model_engine_power_
1	0.001607407	0.50000	0.0000000	1.0000000	0.07084680	0.1666667	0.0000000000	
2	0.001607407	1.00000	0.1547070	1.0000000	0.09001587	0.1666667	0.0107259443	
3	0.063300000	0.22730	0.9832426	0.8889333	0.22185324	0.3333333	0.1314670096	
4	0.063300000	0.90910	0.8972033	0.8889333	0.30379241	0.3412698	0.2081953968	
5	0.063300000	1.00000	0.8753417	0.8480985	0.33768081	0.3874092	0.2400941122	
6	0.063300000	0.40910	0.9549411	1.0000000	0.39444014	0.4351852	0.3149558699	
7	0.063300000	0.95450	0.8651555	0.8633189	0.41509109	0.4637137	0.2816189421	
8	0.063300000	0.04545	1.0000000	0.8889333	0.44394864	0.5000000	0.4254624592	
9	0.063300000	0.86360	0.8630110	0.9444000	0.56962124	0.5179856	0.3951511261	
10	0.063300000	0.59090	0.5378539	1.0000000	0.30930536	0.3898810	0.1781439453	
11	0.063300000	0.18180	0.9672299	0.8333333	0.36972700	0.4074074	0.2989158909	
12	0.063300000	0.63640	0.9414475	0.8333333	0.53803964	0.5686275	0.3898312317	
13	0.063300000	0.54550	0.9423963	0.9421389	0.66431981	0.7291667	0.6596542135	
14	0.063300000	0.45450	0.8764977	0.9110800	0.53870948	0.6083333	0.5512513602	
15	0.063300000	0.04545	1.0000000	1.0000000	0.62152809	0.6666667	0.4907508161	
16	0.063300000	0.31820	0.8847926	0.8610667	0.64710139	0.7083333	0.6825353645	
17	0.063300000	0.13640	0.9009217	0.8889333	0.28225884	0.3333333	0.3240841494	
18	0.063300000	0.72730	0.8847088	0.8454279	0.58934047	0.6439394	0.4575658654	
19	0.063300000	0.13640	0.9608295	0.9722333	0.55288254	0.6041667	0.4793253536	
20	0.063300000	0.36360	0.8687991	0.8855765	0.67219917	0.7450980	0.5199385521	
21	0.063300000	0.04545	1.0000000	0.8889333	0.59193152	0.6666667	0.2092854552	
22	0.063300000	0.40910	0.8863287	0.9444000	0.70074826	0.7592593	0.5197678636	
23	0.063300000	0.81820	0.8926860	0.8707578	0.32869928	0.3787129	0.2834943995	
24	0.001607407	0.66670	0.7941628	1.0000000	0.99883511	0.5555556	0.5392737678	
25	0.001607407	0.50000	0.5505140	0.9444000	1.00000000	0.6666667	0.3275857258	
26	0.063600000	0.09091	0.9815668	1.0000000	0.44394864	0.5000000	0.4631846210	
27	0.063600000	0.09091	1.0000000	0.8333333	0.44394864	0.5000000	0.2092854552	
28	0.063600000	0.81820	0.9116378	0.8333333	0.40479190	0.4682540	0.2839698085	
29	0.063600000	1.00000	0.9041082	0.8333333	0.52843742	0.5248227	0.3702065921	

Figure 3: The clean data for the study

factorial analysis. Figure 8 suggests that the numbers of factors should be three. The interior correlation of the three factors (PA1, PA2, and PA3) are visualized in Figure 9.

Above all, according to the sales over years, a particular car model can be clustered into one of the three classes. One cluster clearly contains those models with total sales more than one million. The distinction of the other two clusters is from the combination of total sales and the time trend, and this distinction generally unseen by eyes only. The size of characteristics can be reduced to four by PCA or to three by FA without loss of important information.

4. EQUILIBRIUM MARKET SHARE COMPUTATION

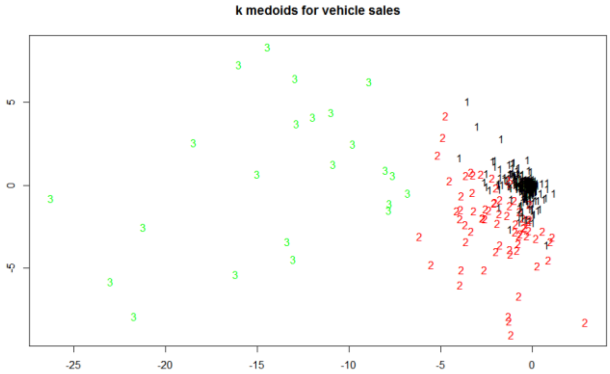
The mathematical model for computing the equilibrium market share are constructed in this section, and this equilibrium should be computed on the scalable framework and at the end of the pipeline, whre the vehicle data is as water flows through the data cleaning and mining described in Section 3.

We first describe the notations used for the least-squares utility estimation. The parameters are N, S_j, x_{ij}, and P_j. N is the numbers of consumers involved in cars-buying. Here N is set at the population containing individuals with age 30 through 39 in UK. The average population between 2001 and 2015-Sep-30 is 61,709,867, and multiplying this number by 13%, which is the percentage of people whose age is between 30 and 39, we get N = 8,022,283. S_j is the market share of each popular vehicle model for j=1, …, 212. x_j is the vector of the characteristic in each popular vehicle types for j = 1, …, 212. P_j is the price for vehicle models j =, 1, …, 212. The output variables are ζ_i, π_{ij}, β_i, α_i, γ_i, where ζ_i is the error of the estimated utility for consumer i = 1, …, 8,022,283. π_{ij} is the choice probability for consumer i to purchase vehicle model j. β_i is the vector of consumer preferences toward the characteristics for consumer i = 1, …, 8,022,283. α_i is a scalar about the consumer's preference for the price P_j for i = 1, ..., 8,022,283. γ_i. is consumer i's maximum utility over all available vehicle models.

The consumer's utility function is

$$u_{ij} = x_j{}'\beta_i - \alpha_i P_j + \zeta_j \tag{1}$$

where u_{ij} is the utility of customer i purchasing vehicle model j.



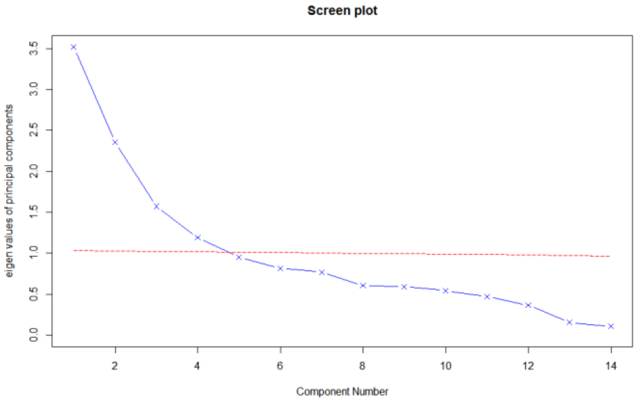


Figure 4: k medoids for vehicle sales

Figure 6: Screen plot for PCA

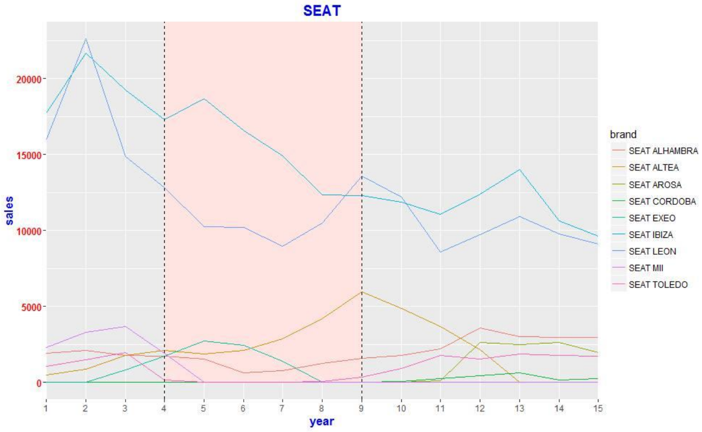
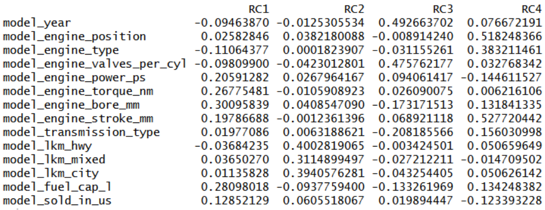


Figure 5: The time trend of SEAT



```
                                 RC1          RC2          RC3          RC4
model_year                -0.09463870 -0.0125305534  0.492663702  0.076672191
model_engine_position      0.02582846  0.0382180088 -0.008914240  0.518248366
model_engine_type         -0.11064377  0.0001823907 -0.031155261  0.383211461
model_engine_valves_per_cyl -0.09809900 -0.0423012801  0.475762177  0.032768342
model_engine_power_ps      0.20591282  0.0267964167  0.094061417 -0.144611527
model_engine_torque_nm     0.26775481 -0.0105908923  0.026090075  0.006216106
model_engine_bore_mm       0.30095839  0.0408547090 -0.173171513  0.131841335
model_engine_stroke_mm     0.19786688 -0.0012361396  0.068921118  0.527720442
model_transmission_type    0.01977086  0.0063188621 -0.208185566  0.156030998
model_lkm_hwy             -0.03684235  0.4002819065 -0.003424501  0.050659649
model_lkm_mixed            0.03650270  0.3114899497 -0.027212211 -0.014709502
model_lkm_city             0.01135828  0.3940576281 -0.043254405  0.050626142
model_fuel_cap_l           0.28098018 -0.0937759400 -0.133261969  0.134248382
model_sold_in_us           0.12852129  0.0605518067  0.019894447 -0.123393228
```

Figure 7: Principal components

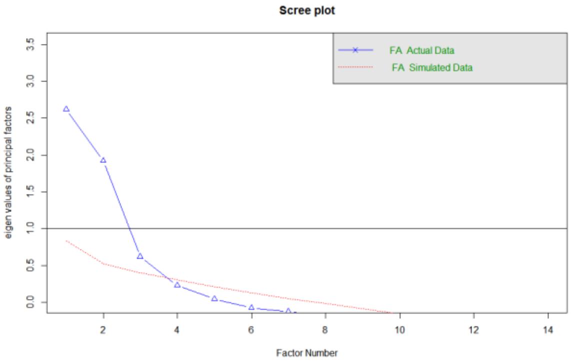


Figure 8: Screen plot for factor analysis

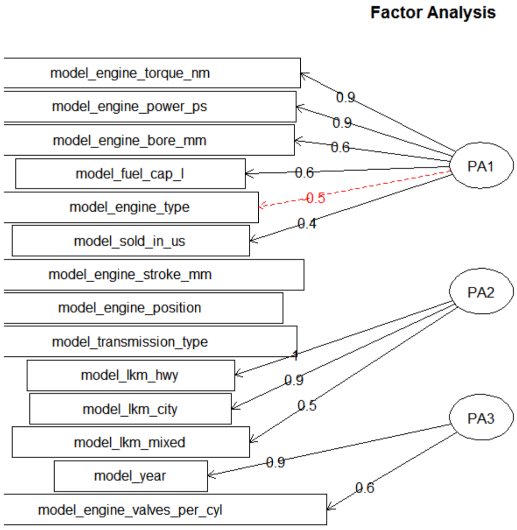


Figure 9. Plot for factor analysis

A quadratic objective function for minimizing the error of estimated utility is desired.

$$Min \quad \zeta^T \zeta \tag{2}$$

In this optimization program, the following constraints are sufficient.

$$(1/N) \times \sum_{i}^{N} \pi_{ij} = S_j \tag{3}$$

$$0 \le \pi_{ij} \perp \gamma_i - (x_j{}' \beta_i - \alpha_i P_j + \xi_j) \ge 0 \tag{4}$$

$$0 \le \gamma_i \perp 1 - \sum_{j}^{J} \pi_{ij} \ge 0 \tag{5}$$

The constraint (1) is the predicted market share for product j. The complementarity constraint (4) implies that the multiplier γ_i satisfies

$$\gamma_j = \max\{0, \max_{1 \le l \le J}(x_j{}' \beta_i - \alpha_i P_l + \zeta_l)\} \tag{6}$$

The complementarity condition indicates that the consumers' choices of the products are in accordance with the maximum ranking(s) of the customer's utilities of the products at the normal prices.

The complementarity constraint (5) means that the following formula will be satisfied.

$$1 - \sum_{j=1}^{J} \pi_{ij} = \pi_{i0}, \sum_{j=0}^{J} \pi_{ij} = 1 \tag{7}$$

The range of π_{i0} is between 0 and 1. $\pi_{i0} = 0$ denotes the consumer buying one of (or several) product j, and $\pi_{i0} = 1$ denotes the consumer choosing to buy nothing.

Suppose now the consumers are facing the market with a specific setting of the price P_j and characteristics x_j for several particular designs of vehicle, the solutions of π_{ij} is interpreted as the optimal strategy for every consumer i according to the estimated payoff structure $u_{ij} = xj^T \beta_i - \alpha_i$. As a result, the equilibrium market share calculated from the total sales divided by total numbers of consumers is computed as in (3).

5. ON-GOING WORK

The plan of model validation is as follows: the data is divided into two groups: training data and testing data. The total data size is 373 in the case study. We draw a size of 212 for the training data and the entire set is the testing data. The training data is used to estimate the structure of utility. The testing data is used for computing the equilibrium market share to be compared with the real one.

REFERENCES

Adomavicius, G., and Tuzhilin, A. (2005) Toward the Next Generation of Recommender Systems: A Survey of the State-ofthe-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering* **17**(6): 734-749.

Brynjolfsson, E., Hu, Y. J., and Smith M. D. (2006) From Niches to Riches: The Anatomy of the Long Tail *Sloan Management Review* **47**(4): 67-71.

Chen, H., Chiang, R. H. L., and Storey, V. C. (2012) Business Intelligence and Analytics: From Big Data to Big Impact. *MIS Quarterly* **36**(4): 1165-1188.

Dean J. and Ghemawat S. (2004) MapReduce: simplified data processing on large cluster. *Communications of the ACM* **50**(1): 107-113.

Einav, L. and Levin, J. (2014) Economics in the age of big data. *Science* **346**(6210): 715-721.

Iskhakov, F., Lee, J., Rust, J., Schjerning B., and Seo, K. (2016) Comment on "constrained optimization approaches to estimation of structural models". *Econometrica* **84**(1): 365-370.

Karau, H., Konwinski, A., Wendell, P., and Zaharia, M. (2015) *Learning Spark: Lightning-Fast Big Data Analysis*. O'Reilly Media.

McAfee, A. and Brynjolfsson, E. (2012) Big Data: The Management revolution. *Spotlight, Harvard Business Review,* 60-69.

McKelvey, R.D. and McLennan, R. (1996) *Computation of equilibria in finite games. In H. M. Amman et al. (ed), Handbook of Computational Economics Volume 1 (Elsevier),* chapter 2, 87-142.

Pang, B., and Lee, L. (2008) Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* **2**(1-2), 1-135.

Pang, J. S., Su, C. L., and Lee, Y. C. (2015) A Constructive Approach to Estimating Pure Characteristics Demand Models with Pricing. *Operations Research* **63**(3): 639-659

Shvachko, K., Kuang, H., Radia, S., and Chansler R. (2010) *IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST),* 1-10.

Sitto, K. and Presser, M. (2015) *Field Guide to Hadoop: An Introduction to Hadoop, Its Ecosystem, and Aligned Technologies*. O'Reilly Media.

Su, C. L., Judd, K. L. (2012) Constrained optimization approaches to estimation of structural models. *Econometrica* **80**(5): 2213-2230.