



**CUSTOMER FLOWS FORECASTS
FOR SHOPS ON KOUBEI.COM
RESEARCH DISSERTATION**

**ZHANG SUPEI
APPLIED COMPUTING
UWTSD**



2017-9-11

Content

1.	Introduction	3
2.	Literature Review	4
2.1.	Data mining application in marketing and customer relationship management	4
2.1.1.	Customer relationship management.....	4
2.1.2.	The use of data mining in CRM.....	6
2.1.3.	Successful Examples.....	8
2.2.	Different data mining method applied with different condition.....	8
2.2.1.	Time Series	8
2.2.2.	Classification	9
2.2.3.		11
3.	CRISP-DM	14
3.1.	CRISP-DM Introduction	14
3.1.1.	Business understanding.....	14
3.1.2.	Data understanding	14
3.1.3.	Data preparation.....	14
3.1.4.	Modeling.....	15
3.1.5.	Evaluation	15
3.1.6.	Deployment.....	15
3.2.	Risk	15
3.3.	Generalizability.....	16
3.4.	Validity	16
4.	Dataset description	17
5.	Pre-processing	20
5.1.	Data cleaning	20
5.1.1.	Translating	20
5.1.2.	Remove irrelevant attributes	20
5.2.	Data loading.....	21
5.2.1.	Secondary Pre-processing.....	21
5.2.2.	Arff file	22
6.	Initial Analysis.....	23
6.1.	Hypothesis.....	23

Customer Flows Forecasts For Shops On Koubei.com

6.2.	Pearson Correlations	23
6.3.	Secondary pre-processing	25
6.4.	Pearson Correlations	26
6.5.	Initial exploration for last two weeks.....	28
7.	Time series	30
7.1.	Secondary pre-processing	30
7.2.	Linear Regression	31
7.3.	Gaussian Processes	33
7.4.	Multilayer Perceptron	34
7.5.	SMOreg.....	34
7.6.	Results comparison	35
8.	Neural Network	37
8.1.	Secondary Pre-processing	37
8.2.	Neural network model.....	38
8.3.	Back Propagation Neural Network	40
8.3.1.	Number of hidden layer	42
8.3.2.	Maximum number of training.....	43
8.3.3.	Result of prediction of BPNN.....	44
9.	Ensembling.....	46
9.1.	Weighted average.....	46
9.2.	Simple average	47
10.	Conclusion	49
10.1.	Time series prediction	49
10.2.	Neural network.....	49
10.3.	Ensembling	50
11.	Reflection	52
11.1.	What I learn.....	52
11.2.	Difficulties	52
11.3.	Expectations.....	52
	Reference	54
	Appendix A: Neural Network code.....	57
	Appendix B: prediction of five algorithms.....	64
	Appendix C: weighted average (not including Multilayer Perceptron).....	66

1. Introduction

Alibaba Group^[1] and Ant Financial Services Group^[2] integrate their resources, and together to create an Internet local life service platform, namely koubei.com^[3]. At June 23, 2015, koubei.com was formally established. Actually, koubei.com, similar with TripAdvisor, is a third-party review site to provide information, and all the information comes from the consumer's comments. Information provided by koubei.com not only includes restaurant, shopping, leisure and entertainment, life services, but also with coupons, community and other related activities.

Sales and customers are crucial for any company. Therefore, customer flow is important for koubei.com. Customer flow could not only help shops to predict the normal flow in order to optimize operations, but also reduce costs, and improve the user experience. Combined with the data mining, I intend to apply CRISP-DM^[4] methodology to analyze the koubei's data, including shop information, customer's payment and review information. My goal is to predict the short-term customer flow at koubei.com.



2. Literature Review

2.1. Data mining application in marketing and customer relationship management

2.1.1. Customer relationship management

In modern society, marketing management is one of the most important elements in the success of modern organizations. ^[5] Marketing and sales are used to create the customer, to keep the customer and to satisfy the customer. With the customer as the focus of its activities, it can be concluded that the number of customers or clients is the key performance indicator ^[6].

Customer relationship management (CRM) ^[7] is derived from the marketing theory. Therefore, CRM is an important approach for each company to manage interaction with current and potential customers. CRM uses data analysis about customers' history with a company and to improve business relationships with customers, specifically focusing on customer retention and ultimately driving sales growth.

Lee^[8] argues that CRM is a customer-centric business strategy that can lead to changes in company's important principles, CRM technology needs and can support business process reengineering (BPR), but BPR is not the result of CRM. Anton's^[9] definition is CRM is the company's important information includes seamless access from the company's internal and external customers, so that through integration of company's telephone system, website and e-mail, customers achieve the purpose of purchasing important products, at mean time. CRM guides incremental sales and cross-selling, and ultimately improves customer loyalty, customer value and customer profitability. Anton's view emphasizes the support role of Computer Telephony Integration (CTI) and Call Center (CC). This view has got the CRM system provider's strong support. Chablo ^[10] tends to define CRM from the point of system integration as a set of methods that integrate all areas of customer engagement and are effectively integrated through people, processes, and technology. Similarly, Imhoff et al.^[11], based on the point of view of system integration, but more emphasis on customer contact management, and CRM is defined as integration method of corporate strategy, organizational structure, culture and customer contact information and technology. SAS ^[12] defines CRM as a process of technology, maximizing the use of customer information to strengthen customer loyalty and retain customer.

The development of information and communication technology led to the development of customer relationship management from theory to practice. At 1990, many American companies began to develop Sales Force Automation (SFA) in order to meet the growing need for competition, and then focused on the development of Customer Service Systems (CSS). Some companies began to merge SFA and CSS systems together with marketing planning and field service, and integrate with Computer Telephony Integration (CTI) to form sale and service call center, which is the embryonic form of CRM. At 1997, Gartner Group ^[13] formally put forward the concept of CRM, accelerated the emergence and development of CRM. After 1998, with the rise of e-commerce, CRM began to be combined with e-commerce. In recent years, many companies in Europe and the United States have used CRM systems, CRM is the change of concept of modern marketing and business operations, and has a wide range of market value and research value.

CRM has four advantages: (1) Improve services, CRM provides customers with active customer care, personalized service based on sales and service history information, and more professional services according to knowledge base. (2) Improve efficiency, with CRM, customers can click to complete a number of business operations, while the degree of service automation increases, making a lot of duplicable work completed by computer system, which makes the quality of enterprise work and marketing efficiency improved. (3) Reduce costs, CRM with modern network technology, relative to the traditional way of marketing, can greatly reduce the cost of marketing operations, coupled with the ability to accurately find customers, and achieve online information exchange, which can greatly develop one-to-one marketing and other new forms of business. (4) Expand sales, CRM makes the accuracy of sales, and success rate grow, customer satisfaction, sales expansion will become inevitable.

CRM is focused on the customer requirement. The customer-centric marketing strategy consists of three aspects ^[14]. Firstly, customer identification make companies find valuable customers; Secondly, customer behavior analysis explore and guide customer needs, so it maintains, retain customers; Thirdly, according to the previous analysis of the results of the customer, appropriate decisions are made in order to drive production. To achieve this, modern enterprises must establish an integrated and efficient customer relationship management system, in order to fully manage the enterprise customer resources, and customer needs to guide the overall business activities. KDD or data mining can be applied to the above three phases. based on unified customer data, using online analytical processing (OLAP), statistics, especially data mining analysis, CRM could accurately understand customer behavior, expectations, needs, history and business integrated relationship, which will increase the value of information analysis, and provide support for strategic decision-making. It can be said that if there is no successful customer information analysis and management, CRM value can not really work.

Implementation of customer relationship management is inseparable from customer data collection and storage. With business process automation, many enterprises have produced a few GB or even TB of customer data every day, such as Wal-Mart ^[15] created a capacity of 43TB customer database records with each transaction and business income and expenditure, etc. However, faced with such a large amount of data, the traditional analytical methods encountered difficulties, if there is no a strong data analysis tool, companies will fall into situation "data rich, but the information poor," a large number of customer data hidden useful information, which will be submerged, the goal of customer relationship management is difficult to really achieve. The emergence of data mining technology meets this need.

In my case, my target customer is those consuming at retailers, such as restaurant, groceries at different cities. Retailers want to improve efficiency, and build a competitive advantage, so they must combine with its operating characteristics, make full use of modern customer relationship management ideas and technology. Retailers sell many types of goods, operate complex structures, the general business includes tens of thousands of goods, furtherly some large retail stores sell up to hundreds of thousands of types of goods. In this way, every day a large number of goods and transactions information needs to be processed. At the meantime, this information is the main basis of unified management, in particular, of unified distribution. The vast majority of customers are single consumers, which are large, widely distributed, complex structure, in accordance with the

general method; it is difficult to carry out one to one marketing. Retail's customers are also sporadic customers, they buy a small number of goods, but the purchase frequency is high, some of which customers will periodically patronize some fixed retail stores. Discovering these customers, consolidating and developing relationships with them is an important part of customer relationship management.

Retail stores generally take clustered distribution, and many retail stores are centrally located in the city's business district. The clustered distribution of retail stores can produce aggregate effects, maximizing passenger flow and increasing store traffic. Nevertheless, the store's clustered distribution also increases the competition between stores and stores. In order to attract customers, the store often adapt the price war, promotion war, which increase the cost of the store sales. However, through customer relationship management, training their loyal customers is an effective way to replace the price and promotional competition. Not only do retailers achieve the purpose of long-term retention of customers, while the cost should be much lower than other competition. In fact, the early commodity distribution store in the residential area has adopted a customer relationship management method, the grocery store owner will record habits and requirements of frequent shopping customers, when customer come to buy once again, they provide targeted services. This kind of personalized service was the way of business and survival of many grocery stores. Latterly due to increase in the number of goods and customers, the amount of information is also greatly increased, can retailers no longer record each customer to serve. The CRM combined with data mining provides technical support for one-to-one marketing in modern retailing, enabling large retail stores to provide personalized service for every customer as a grocery store owner.

2.1.2. The use of data mining in CRM

Data mining (DM) is extracting information from a large number of incomplete, noisy, fuzzy and random data, which is not known in advance, but is potentially useful information and knowledge. In fact, data mining technology is application-oriented. It is not only a simple search query call for a particular database, but also micro- or macro-statistics, analysis, synthesis and reasoning to guide the solution of practical problems, and attempt to discover the interrelationships between events, even use existing data to forecast future activities. In my case, existing customer and transaction data from database is analyzed to predict the sale trend. All the knowledge found is relative, there are specific prerequisites and constraints for specific fields, but knowledge also is able to be easily understood by the user.

Data mining, also known as knowledge discovery in the database (KDD), refers to advanced processing extracting a credible, novel, useful and understandable model from a large number of data. Actually "Model" can be seen as the embryonic form of knowledge, after verification, improved as knowledge. Advanced processing is a multi-steps process, multi steps interact between each other, are repeated adjustment, and format a rising spiral. Since data mining is the core step of KDD, a hidden pattern is found, so many people think that the two are equivalent from the perspective of pattern processing.

The term of KDD was formally formed at the first KDD International Conference in Detroit, USA, in August 1989^[16]. The first international conference on knowledge discovery and data mining was held in Canada in 1995^[17]. The data mining came quickly as the data in the database was likened

to the deposit. Since 1995, DM has become a hot research field, some excellent journals have come such as: such as "Data Mining and Knowledge Discovery ^[18]", "Artificial Intelligence Review ^[19]".

Data mining research is almost synchronous with its application. Since the 1990s, many data mining software tools have been developed. These data mining tools can be divided into two categories according to their applicable scope: dedicated data mining tools and general data mining tools. Dedicated data mining tools are for a specific area of the problem to provide solutions, and embedded in the application system, when the algorithm are designed to take full account of the data, particularity of demand. At present, the main applications of such solutions are: KDI (mainly for retail), Optiong & Choice (mainly for the insurance industry), HNC (fraud detection), Unica Modell (mainly for marketing). On the other hand, the general data mining tool does not distinguish the meaning of specific data, takes use of common data mining algorithms, deals with common data types. Typical generic data mining tools include IBM ^[20] Intelligent Miner, Clementine, Enterprise Miner for SAS ^[21], and Oracel ^[22] Darwin.

Data mining plays a key role in customer relationship management. With the support of data mining technology, CRM concept and goal can be realized. The customer data collected by the enterprise's database accumulates massive amounts of data about customer behavior and characteristics, for example, in my koubei case, the company just provide one and half year data, which records are up to 70,000,000. Moreover, enterprises could use data mining to discover the hidden and valuable customer knowledge from this massive amount of data for better insight and understanding of its customers, and develop effective marketing and customer service strategy to provide an important basis for decision-making. Data mining is focused on the lifecycle of customer relationship. In fact, each life stage about customer is important for marketing and customer relationship management, for example, moving is a significant event.

Specificly, CRM combined with data mining could increase business profit in every stage of the relationship: (1) obtain new customers through data mining. The first step in CRM is to identify potential customers and then turn potential customers into real customers. Data mining can help customers segment and identify potential customers. (2) improve customer value. Data mining supports customer profitability analysis, and predicts customer profitability changes to enhance customer profitability, in addition supports customer purchase behavior model analysis, customer segmentation, provides more targeted personalized service, besides guides effective cross-selling, retailers could provide new products and services to existing customers to achieve the purchase of recommended and incremental sales. (3) Customer retention. This section includes customer loyalty analysis and customer loss analysis, through data mining, retailing and more companies could analyze customer transaction behavior, warn customer abnormal behavior, and put forward the corresponding countermeasures and suggestions.

With major trends emerge, such a shift in the economy toward E-commerce, online to offline (O2O) is subpart of E-commerce, actually O2O is a phrase that is used in digital marketing to describe systems enticing consumers within a digital environment to make purchases of goods or services from physical businesses.^[23] O2O is a hot research issue at China, because lots of physical companies convert their businesses from offline to online, such as restaurants, clothing shops, and services, in fact mostly products and services could be bought or ordered online.

2.1.3. Successful Examples

The use of data mining tools for customer relationship management has made many companies get a valuable return. Examples of US West Telecom and British Bass Export companies are brilliant examples [24].

US WEST, as one of the largest long-distance telecommunications companies in the United States, has more than 20 million customers. The company is currently using the SAS Research Institute's Enterprise Excavator Software to further enhance its successful target market strategy. Western Telecom took use of sales activity management software, along with SAS's corporate excavators, to enable marketing specialists to plan, execute and evaluate sales activities that are targeted. Combining data mining tools with sales activity management software eliminates the burden of salespeople grading all customer data (which can be time-consuming) and reduces the human error caused by manual intervention. As a result, the marketing cycle is greatly shortened, and because of the market for more detailed and highly targeted division, the enterprise get a higher marketing return on investment.

Bass Export is the UK's largest beer exporter. Bass export currently has nine breweries, and more than 80 overseas markets to sell more than 60 kinds of beer products. The company realized that relying solely on a large number of brands and good service was not enough to keep the company in the UK beer market in the dominant position, so the choice of IBM's intelligent excavator as its data mining business solutions. The company's IS department manager explained: "We have to complete 23,000 customer orders per week. For us, understand every customer's buying habits, the profitability of each account, and we sold each one Brand profitability is essential. "

2.2. Different data mining method applied with different condition

Tasks of data mining are to find potential and useful patterns from the data set. The model can be divided into two categories: a predictive model, which refers to predict an unknown result according to the value of the model; the descriptive model, which refers to make a normative description, and find the general characteristics of the data.

In recent years, the research of data mining has been changed according to the different data types. For example, the transaction records in the traditional database has changed from simple numerical data to complex data such as tense, space, text and multimedia. The structured data is transformed into semi-structured data. Static data changes to dynamic data and so on. For different data types, the corresponding data mining theory, methods and application are different, which are difficulties at the field of data mining research.

2.2.1. Time Series

Time series is a kind of very common and time-related high-dimensional data, which is also the main research object in the field of data mining. It is widely applied in finance, medicine, meteorology, industrial and engineering areas, such as stocks, atmospheric temperature and engine performance parameters.

Similarly, some time-independent objects can be transformed into time series data, such as gene data and object shapes [2]. In recent years, with the development of socio-economic and information technology, the amount of data in the sequence grows faster greatly. Therefore, how to

mine from a large number of time series data and to find information and knowledge is the main research.

Time series data is usually high dimensional, its production process is vulnerable to environmental factors and noise. In practice, it is usually necessary to carry out local feature extraction or global feature decomposition for time series data, in order to reduce the dimension of the original time series, at the meantime the similarity measure methods are combined to improve efficiency.

Time series data are growing as time goes, and the high dimension, dynamic and uncertainty of data reduce efficiency. If the data mining algorithm is used directly to extract information and discover the knowledge on the time series data, the time cost and space resource consumption are not only high, but the accuracy and reliability of the final results can not be guaranteed. Therefore, how to improve the performance of time series data mining technology is the focus and difficulty. In general, improving the current algorithm and data preprocessing are two important means to improve the performance. The former is an improvement of the design of the specific algorithm, it depends on the specific data structure selection and algorithm model, the specific time series data can show better performance, such as clustering algorithm, classification algorithm, association rule algorithm. The latter is the transformation of the high-dimensional data, namely it transforms into the data object under the low-dimensional space ^[25], and retains most of the characteristic information of the original time series data, so that the data mining technology in the low-dimensional space plays a better effect, usually referred to as this time series data transformation process for dimensionality reduction or feature representation.

At present, the research on time series feature representation is paid more and more attention. Some series data mining experts usually extract the characteristics of time series, and then apply it to time series data mining ^[26]. There are many time series feature representation methods, such as piecewise linear representation, segmented aggregation approximation, symbolic representation method, representation method based on domain transform, singular value decomposition and model-based representation method.

In 1993, Agrawal et al. ^[27] proposed for the first time a feature representation method that transforms time series of the time domain into frequency using discrete Fourier transform and applies it to time series similarity searches. Since then, the feature representation and similarity measure of time series have gradually become a hot issue in the study of time series data mining. Many famous scholars and experts have been involved in research, in which Pazzani and Agrawal research group at IBM is an early exhibition research institutions, and Keogh ^[28] at University of California Riverside led the research team, which is a very strong and creative team to study time series data mining.

2.2.2. Classification

There are several types in the classifier: mathematical statistics method, artificial intelligence, neural network and so on. According to the classification algorithm, research can be divided into the following categories: decision tree algorithm, Bayes algorithm, association rules, neural network, K-nearest neighbor algorithm, genetic algorithm, rough set and so on. Decision tree and association rules are typical artificial intelligence methods. On the other hand, the statistical methods mainly

include Bayesian classification algorithm and K-nearest neighbor algorithm. BP neural network is mainly algorithm at ANN ^[29]. The model of the BP algorithm is the forward propagation of the signal and the back propagation of the error, which is a nonlinear continuous transfer function.

(1)Decision Tree

Decision Tree is a directed acyclic graph (DAG) ^[30]. The decision tree method takes use of the information gain in the information theory to find the attribute field with the largest amount of information in the database, establish a node of the decision tree, and then establish the tree branch according to the different value of the attribute field, repeat in each branch subset, then build the lower nodes and branches of a tree. The concrete process of constructing the decision tree is to find the initial division first, and the whole training set is a set of decision trees. Each training record must be a good class to determine which Field is the best class index. The general practice is to try all the attributes of the domain, quantify the division of each domain to calculate the best of a split. The standard for quantification is to calculate the diversity of each division. Secondly, it needs to repeat the first step until the records within each leaf node belong to the same class and grow to a complete tree.

Among the algorithms used in the current decision tree, the most influential one is the ID3 algorithm Quinlan ^[31] proposed in 1986, and he proposed the use of information gain as a selection criterion for attributes so that the maximum category information on the test record can be obtained when testing for each non-leaf node. ID3 always selects the attribute with the highest information gain as the test attribute of the current node. The specific method is to detect all the attributes, select the attribute with the largest gain of the information to generate the decision tree node, set up the branch from the different values of the attribute, and then recursively call the method to establish the branch of the decision tree node,

C4.5 algorithm overcomes some shortcomings of ID3 algorithm, which is an extension and improvement of ID3 algorithm. (1) The ID3 algorithm uses the information gain to select the test attribute, while the C4.5 algorithm uses the information gain rate Select, which is more accurate, and will not bias the attributes of more value.(2) pruning operations in the process of spanning trees;(3) discretizing continuous attributes;(4) dealing with the missing data.

The ID3, C4.5 algorithm is the earliest decision tree classification algorithm, but it is only valid for smaller training sample sets, until all the subsets contain only the data of the same category. In order to construct the decision tree with ID3 algorithm, the classification efficiency is not high. Olaru ^[32] proposed a fuzzy decision tree algorithm based on fuzzy method is proposed, which greatly improves the correct rate and inductive ability of tree.

(2)Artificial neural network

The development of artificial neural network (ANN) has undergone a tortuous process. Its development process can be divided into five periods: 1943 -1969 is the first craze; 1969?-1982 is trough; 1982 – 1987 is the beginning of the second boom; The late nineties of twentieth century is the second craze; the twenty-first century so far falls into the sustainable development stage.

The earliest study of ANN dates back to the Frued ^[33] period at the nineteenth century and his

neuro analysis has established a basic study of the nervous system. In 1943, the American mathematician Pitts and neuropsychologist McCulloch ^[34] jointly published a forward-looking paper, they first proposed the concept of neurons, and put forward the famous neural network model, known as the M-P model. In 1949, the psychologist Hebb ^[35] proposed the famous "Hebb learning rule (hypothesis of variability in the connection of neurons)", which is still being followed by most learning machines. In 1958, Rosenblatt ^[36] proposed the concept of Perceptron, which was considered the earliest neural network model. In 1962, Widrow and Hoff ^[37] developed a network model called Adaline, which is also called the Widrow-Hoff algorithm. In 1987, First Annual International Conference on Neural Networks was held in San Diego, USA, where the International Neural Network Society (INNS) was established. Latterly a variety of neural network meetings have been held, the new upsurge of neural networks has been formed. Since 1995, the neural network has entered a period of stable and sustained development, and a lot of researches have been made, such as Xu ^[38] proposed Ying-Yang machine theory.

At present, there have been many kinds of artificial neural network model, which can be classified from multiple methods. According to the topology of artificial neural network, it can be divided into feedforward neural network and feedback neural network. According to the performance of artificial neural network, it can be divided into deterministic, random, continuous, and discrete neural network. According to learning method, it can also be divided into supervised and unsupervised ANN.

The application of artificial neural network in data mining has many advantages, the most important merit is the excellent parallel processing and strong self-organization learning ability, which is widely used in various areas in recent years. Moreover, pattern recognition is one of the most important advantages, which is good at solving the classification problem. There are many kinds of popular artificial neural network models, such as BP neural network, RBF neural network, Hopfield network, probabilistic neural network (PNN) and self-organizing feature mapping network (SOM).

2.2.3. Ensembling learning

Model ensembling is a very powerful technique to increase accuracy on a variety of data mining tasks.

The traditional method of machine learning is to firstly construct a functional hypothesis space for the problem. For the classification problem, the purpose of the optimization is to obtain the classifier h closest to the actual function f ^[39]. Multi-models ensembling learning builds and combines multiple models to complete the learning task, the process is to produce a group of individual learners, and then use a strategy to combine them ^[40]. The accuracy, generalization and stability of the models system will be significantly improved by multi-model ensembling learning. Dietterich ^[41] argues that multimodal ensembling learning can significantly improve the effect due to statistics, calculation, and expression.

Many of the theoretical studies of multi-model ensembling learning are focused on weak learners, although theoretically, the use of weak learners could result in better performance, but in practice for various considerations, such as the use of less individual learners, or the reuse of some experienced learners, people tend to use a relatively strong learner.

Individual learners are usually generated from the training set using existing learning algorithms, such as the CART decision tree algorithm or the BP neural network algorithm, where the multimodal ensembling contains only the same type of individual learner, such as "decision tree integration" is the decision tree, "neural network integration" is all the neural network, such a multi-model ensembling is "homogeneous", this individual learner is also known as "base learning device." Multimodel ensembling learning can also combine a number of different types of individual learners, such as the simultaneous integration of BP neural networks and CART decision trees, such multimodel ensembling learning is "heterogeneous", correspondingly, individual learners are generally not known as the base learners, often referred to as "component learners" or directly called individual learners ^[42].

Multimodel ensembling learning algorithms include Boosting, Bagging, and Stacking and so on. It is in the Boosting algorithm that there is a strong dependency between the individual learners, which is the serial generation. During the training process, the proportions of the training data are adjusted, such as AdaBoost and GBDT. By the contrary, there is no strong dependency between the individual learners in the Bagging algorithm, and the parallelization is generated, such as Random Forest. The basic idea behind stacked generalization is to use a pool of base classifiers, then using another classifier to combine their predictions, with the aim of reducing the generalization error. Stacking algorithm is divided into two fold, the first stage is to use different algorithms to generate multiple initial weak learner, at the same time, according to the output of a number of learners a data set with the same size of the original data set is produced, and then use the new data set and a new secondary learning algorithm form the second stage classifier.

Multi-model ensembling learning is widely used. American scholar Hansen^[43] takes use of majority voting method combined with neural network learning method in the handwriting recognition, and ultimately achieves very good results, compared with the traditional one of the best neural network algorithm, there are 20% to 25% improvement. Cherkau^[44] studied the multi-model ensembling of simple averaging method and used it to integrate multi-layer feedforward neural networks with different hidden layers. They used this method instead of NASA's Gauss classifier to analyze the collected synthetic aperture radar images of Venus from the Magellan space detector, in one specifical respect, the ensembling learning reached the expert level of planetary geology. Professor Zhou Zhihua ^[45] at Nanjing University and Microsoft China Research Institute, Carnegie Mellon University, applied the multi-model ensembling of neural network for face recognition, and achieved a very high precision.

Google's DeepMind team developed the AI algorithm: AlphaGo in the Go game has made a major breakthrough. In October 2015 it first defeated the top European professional players, and then in March 2016 to 4: 1 to beat the top international players Li Shishi, causing global public opinion commotion. This has a milestone in the field of artificial intelligence. Go is much more complex than chess, and it is difficult to assess the impact of a step on the overall situation. The DeepMind team's paper "Mastering the game of Go with deep neural networks and tree search" ^[46] on "Nature" on January, 27, 2016, says that based on a variety of existing innovations algorithms, they combined the Monte Carlo Tree Search (MCTS) algorithm with deep neural networks, then applied supervised learning of human chess and self-matching simulation to train the learner, take use of Value Networks to assess the location, and Policy Network to select the footwork, and

ultimately to achieve a complete improvement beyond the existing Go program.

Combined with Customer Relationship Management, the dataset could be analyzed to predict the customer amount every day. According to the data attribute characteristics, time series and classification are mature and developed algorithms to meet the requirement.

3. CRISP-DM

3.1. CRISP-DM Introduction

I decide to adopt CRISP-DM ^[47] (cross-industry standard process for data mining). CRISP-DM defines the process of asking questions, analyzing problems and solving problems in analytical applications. The CRISP-DM model is now a de facto industry standard and provides a comprehensive overview of the data mining lifecycle, which are commonly applied to large-scale project management. The CRISP-DM model takes a leading position in the various KDD process models, which accounts for nearly 60% share ^[48].

CRISP-DM breaks the process of data mining into six major phases, business understanding, data understanding, Data preparation, modeling, evaluation, deployment.

3.1.1. Business understanding

In this initial phase, the requirements and the ultimate goal from a business perspective need to be explained combined these with the background, then the real problem is converted to a specifically data mining one. To achieve the goal, an initial plan is designed. For data mining to be successful, the business goal should be well defined and directed towards particular effort that are amenable to analysis using available data. A data mining business goal can usually be expressed in terms of something measurable such as incremental revenue, response rate, order size.

In this case, the final aim is assisting shops manage operation to improve the customer relationship, which fatherly cuts cost and make more profits. Specifically, the particular goal is predicting short-term sales for shops on Koubei, therefore shops can manage supply and demand response according to the predicted customer flow. The key tasks include data understanding, data processing, identifying important factors, modeling, evaluation.

3.1.2. Data understanding

The data understanding starts with an initial data collection and proceeds with activities in order to get familiar with the data. At first, data quality problems, meanings of each attribute, relationships between the attributes, recongnization of key attribute are the main content.

Luckily, Koubei.com provides me the dataset, including shop information, user payment, and user history view. Each table has particular attributes, I need to understand meaning of each attribute, and remove the invalid ones.

3.1.3. Data preparation

The data preparation phase covers all activities to construct the final dataset from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Data preparation is the most time-consuming part of data mining project. Some data preparation is required to fix problems with source data, but much of it is designed to enhance information content of data. Better data means better models. Typically, data from a variety of sources must be combined to form a customer signature with one record per customer and a large number fields to capture everything of interest about them. Because the source data is usually not the customer level, building customer signature requires many transformations. Transactions must

be summarized by useful ways. Trends in time series might be captured as slopes or differences. Tasks include table, record, and attribute selection as well as transformation and cleaning of data for modeling tools.

3.1.4. Modeling

In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often needed. After the model has been deployed, its purpose is to score new data where the probability of response, class, or value to be estimated is unknown. Applying a model to new data is called scoring. The data to be scored must contain all input variables required by the model along with a unique identifier for each row.

3.1.5. Evaluation

Before proceeding to final deployment of the model, it is important to more thoroughly evaluate the model, and review the steps executed to construct the model, to be certain it properly achieves the business objectives. A key objective is to determine if there is some important business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

3.1.6. Deployment

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that is useful to the customer. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data scoring or data mining process. In many cases it will be the customer, not the data analyst, who will carry out the deployment steps. Even if the analyst deploys the model it is important for the customer to understand up front the actions which will need to be carried out in order to actually make use of the created models.

3.2. Risk

The data mining is a way of learning from the past in order to make better decisions in the future. There exist risks at learning process, which include all factors, or process leading to unexpected results, and there are two undesirable outcomes of learning process:

Learning things that are not true.

Learning things that are true, but not useful.

Learning things that are not true is more dangerous than learning things that are useless because business decisions may be made based on incorrect information. Data mining results often seem reliable because they are based on actual data processed in a seemingly scientific manner. This appearance of reliability can be deceiving. The data may be incorrect or not relevant to the question at hand. The patterns discovered may reflect past business decisions or nothing at all. Data transformation such as summarization may have destroyed or hidden important information.

The most careful and painstaking analysis, using the most sophisticated techniques, yields

incorrect results when the data analyzed is incorrect or simply not relevant, as popular sayings, “garbage in, garbage out”. Particularly, there are three main results to be avoided. Firstly, patterns may not represent any underlying rule, figures do not lie, but liars do figure. When it comes to finding patterns in data, figures do not have to actually lie in order to suggest things that are not true. The challenge for me at this case is to figure out which are useful and which are not. Secondly, the model set may not reflect the relevant population. The model set is the data used to create a data-mining model, and it describes what happened in the past, the model can only be as good as the data used to create it. Therefore, the model set must reflect the population that the model is meant to describe. A sample that does not properly reflect the population being scored is biased. A biased model set can lead to learning things that are not true, furtherly, the resulting model is also biased. However, biases can be hard to avoid. In this case, because the operation time is different according to the each shop, there are so many missing values at some periods, so records with no missing values reflect a different population from records with missing values. Thirdly, data may be at the wrong level. Failure to aggregate to the appropriate level can also lead to confusion.

Learning things that are true, but not useful. Learning things that are not useful is more common. Firstly, learning things that are already known. As we know, data mining should provide new information, however, many patterns in data are already known. Learning things that are already known dose serve one useful purpose, which demonstrate that data mining are working and data is reasonably accurate, even if not helpful. In this case, customer attrition may show that a strong predictor of customer leaving is the way they are acquired, maybe it is too late to go back and change that for existing customers, but future attrition could be reduced by changing.

3.3. Generalizability

In this case, the data mining model could be applied on other general dataset, which is also the goal of this project. Firstly, koubei.com provides the information of shops and each payment records, models based on the original dataset also could process similar data of other shops and more records. Secondly, the data mining techniques have good portability, including time series, neural network, and classification. Though each method may have different constraints, and formats, a model combined with variable techniques could deal with the original data.

3.4. Validity

The key to dealing with a difficult problem is recongnizing a goal, then breaking down into many small goals or processes. In this case the main goal is predicting a short-term customer flow, many data mining techniques are often used for this purpose, for example, BP neural network algorithm success on the classification of medical sets, like "whether suffering from heart disease", Naive Bayesian are usually applied to forecast the stock price, decision trees can quickly classify data sets and improve their ability to classify data correctly, K nearest neighbor classification algorithm is most suitable for application in classification and regression model. These mature algorithms have been applied on thousands of projects, the results imply that appropriate models can successful predict the future trend.

4. Dataset description

There are 3 datasets, namely shop_info, user_pay, user_view, every dataset has its own structure and meanings. The most important dataset is user_pay recording each payment for each shop, unfortunately there is no payment amount due to commercial secret. shop_info provides the basic shop information, such as location, and category. User_view provides the shop history reviewing, which may have a positive or negative relationship with the sales.

Tab 4.1 General description of raw data

Dataset	Amount of Record	Amount of Attributes	time
shop_info	2000	10	--
user_pay	69674110	3	2015-06-26 06:00--2016-10-31 23:00
user_view	2097510	3	2016-02-01 00:00--2016-10-31 23:00

(1)Shop_info

Shop_info is about features of shops. There are 2000 shops with total 10 attributes, those shops are chosen randomly from different cities, and however the close shops share the same location_id. Though I cannot obtain the payment of each transaction, the shop_info provides the average payment per person. Besides, the score and amount of comments are pivotal factors, some hypotheses could be made about the relationship between customer flow and score. Shop level and category may have an influence on customer flow, in common sense, the higher level the shop is, the more customers it attracts.

Tab 4.2 shop_information

shop_information		
Field	Type	Description
shop_id	numerical	Restaurant_id
city_name	string	City name
location_id	numerical	Location numbering, shops with similar location have same number, like postcode
per_pay	numerical	Average payment per person, the higher this value is, the higher the consumption is
score	numerical	Score from users, the higher the score is, the more satisfaction is
comment_cnt	numerical	Comments from users, the higher the value is, the more comments users make
shop_level	numerical	Restaurant level, the higher the value is, the higher level the shop has
cate_1_name	string	First Category
cate_2_name	string	Secondary Category
cate_3_name	string	Third Category

(2)User_pay

User_pay records that users pay at the offline shops, not delivery, sometimes users pay after viewing, and on the other hand, users directly pay without viewing. It is difficult and nonsense to make a model on each user's purchase, the key is shop_id, which is the goal of the data mining project. The time_stamp is processed, so the specific hour is integer, so is the time_stamp in User_view.

Tab 4.3 User pay history

user_pay		
Field	Sample	Description
user_id	1	Customer id
shop_id	1	Shop_id
time_stamp	2015-10-10 11:00:00	Pay time

(3)User_view

User_view has the same attributes with user_pay, therefore the only difference is time. When the user views the detail of a shop (click to the shop detail page), the database will make once record. This sector has similar meaning with the score from shop_info, but there is no particular score at user_view. On the other hand, the score from shop_info is made at 2016-10-31, which means the relationship between score and early customer flow may be weak.

Tab 4.4 User view history

user_view		
Field	Sample	Description
user_id	1	User_id
shop_id	1	Shop_id
time_stamp	2015-10-10 10:00:00	View time

5. Pre-processing

5.1. Data cleaning

5.1.1. Translating

In the original data set, the city name and category at shop_info is Chinese, so I take use of the function of find and palce at EXCEL to translate Chinese to English.

Tab. 5.1 Translation from Chinese to English

shop_id	1	1
city	湖州	Huzhou
location	885	885
per_pay	8	8
score	4	4
comment_cnt	12	12
shop_level	2	2
cate_1_name	美食	food
cate_2_name	休 闲 茶 饮	drink
cate_3_name	饮品/甜点	

5.1.2. Remove irrelevant attributes

the shop_info.file has 10 attributes, but there are too many missing values at the cate_3_name, namely Third Category, there are up to 50% missing values, on the other hand, the cate_1_name and cate_2_name provide the first and second category, which provides enough information , so I decides to delete this cate_3_name. There are 9 attributes left.

Tab 5.2 New shop_info

shop_id
city
location
per_pay
score
comment_cnt
shop_level
cate_1_name
cate_2_name

5.2. Data loading

The records of the original User_pay dataset are up to 70,000,000, most of them are disorganized, so I take use of Tableau ^[49] to unfold the data according to the date.

Tab 5.3 samples of cleaning data

Date/shop_no	1	2	3	4	5	6	7
2015/6/26							
2015/6/27							
2015/6/28							1
2015/6/29							
2015/6/30							
2015/7/1							

5.2.1. Secondary Pre-processing

Due to different opening time of shops on the Koubei, so there are lots of missing values at beginning, especially at 2015, I fill the values as 0, because it represents actual none.

Tab 5.4 samples of filling missing values

Date/shop_no	1	2	3	4	5	6	7
2015/6/26	0	0	0	0	0	0	0
2015/6/27	0	0	0	0	0	0	0
2015/6/28	0	0	0	0	0	0	1
2015/6/29	0	0	0	0	0	0	0
2015/6/30	0	0	0	0	0	0	0
2015/7/1	0	0	0	0	0	0	0

5.2.2. Arff.file

The next step is creating the .arff file, so I add the @RELATION, @ATTRIBUTE and @DATA. Then I modify the attribute name to represent specific meaning with the help of Nik, my supervisor.

```
@RELATION  dailysale
```

```
@ATTRIBUTE saledate DATE "dd/mm/yyyy"
```

```
@ATTRIBUTE shop1 numeric
```

```
@ATTRIBUTE shop2 numeric
```

```
@ATTRIBUTE shop3 numeric
```

```
.....
```

```
@ATTRIBUTE shop1998 numeric
```

```
@ATTRIBUTE shop1999 numeric
```

```
@ATTRIBUTE shop2000 numeric
```

```
@DATA
```

```
06/03/2015,165,117,0,0,0,58,195,....
```

```
07/03/2015,191,82,0,0,22,31,81,.....
```

```
08/03/2015,136,103,0,0,18,43,51,.....
```

6. Initial Analysis

According to the definition of koubei.com, customer flow is the quantity of customer payment at the period, at this case, the payment amount is confidential so it is not public, and the period means one day. Based on the definition, the original dataset have each record, it is unrealistic and unnecessary to predict customer flow at one hour or each customer, I should focus on each shop. The dataset after secondary preprocessing could be used to explore the initial analysis.

6.1. Hypothesis

Before deep mining the patterns, it is better to make the hypotheses after the business and data understanding. Those hypotheses are just assumptions at the beginning.

H1 Six attributes have an effect on customer flow

H1.a Location has an effect on customer flow

H1.b Customer score has an effect on customer flow

H1.c Quantity of comments has an effect on customer flow

H1.d Average payment has an effect on customer flow\

H1.e Shop level has an effect on customer flow

H1.f Shop category has an effect on customer flow

At the same time, date, the most important factor, may reflect some fluctuation at customer flow, furtherly the prediction could be made according to the recent or ultimate data.

H2 customer flow fluctuates on time

6.2. Pearson Correlations

In order to explore the relationship among those attributes, I apply the Pearson correlations to seven attributes, namely total customer flow for each shop, average payment, score, comment amount, shop level, location, shop id.

As following, when correlation is significant at the 0.05 level (2-tailed), there are 8 relationships. Total customer flow has a relationship with average payment, score, comment amount, shop level, however total customer flow just has weak relationship with average payment (0.182), score (0.069), shop level (0.290), and a strong relationship with comment amount (0.446), therefore I could pay more attention to these four attributes, by the contrast, the location, which I thought was pivotal, could be reduced the focus. Therefore, H1.b, H1.c, H1.d, H1.e are suggested to be ture. Besides, what's interesting is other relationship among those attributes, including per person payment amount is related to quantity of comments and shop level, which are also positive, customer score has weak positive relationship with shop level, quantity of comments has strongly positive relationship with shop level.

Tab 6.1 Correlations among attributes

Correlations								
		total customer flow	per_pay	score	comment_cnt	shop_level	location	shop_id
	Pearson Correlation	1	.182**	.069**	.446**	.290**	0.043	0.026
	Sig. (2-tailed)		0	0.005	0	0	0.054	0.237
	N	128000	2000	1709	1709	2000	2000	2000
	Pearson Correlation	.182**	1	-0.014	.176**	.268**	0.043	-.048*
	Sig. (2-tailed)	0		0.564	0	0	0.056	0.032
	N	2000	2000	1709	1709	2000	2000	2000
	Pearson Correlation	.069**	-0.014	1	0.041	.104**	0.023	0.043
	Sig. (2-tailed)	0.005	0.564		0.09	0	0.339	0.075
	N	1709	1709	1709	1709	1709	1709	1709
	Pearson Correlation	.446**	.176**	0.041	1	.547**	-0.004	0.011
	Sig. (2-tailed)	0	0	0.09		0	0.87	0.656
	N	1709	1709	1709	1709	1709	1709	1709
	Pearson Correlation	.290**	.268**	.104**	.547**	1	0.012	0.01
	Sig. (2-tailed)	0	0	0	0		0.606	0.665
	N	2000	2000	1709	1709	2000	2000	2000
	Pearson Correlation	0.043	0.043	0.023	-0.004	0.012	1	0.003
	Sig. (2-tailed)	0.054	0.056	0.339	0.87	0.606		0.89
	N	2000	2000	1709	1709	2000	2000	2000
	Pearson Correlation	0.026	-.048*	0.043	0.011	0.01	0.003	1
	Sig. (2-tailed)	0.237	0.032	0.075	0.656	0.665	0.89	
	N	2000	2000	1709	1709	2000	2000	2000
** Correlation is significant at the 0.01 level (2-tailed).								
* Correlation is significant at the 0.05 level (2-tailed).								

What impresses me is the location has no even weak relationship with any other attributes. Sirgy^[50] argues that geographic is one of several determinants of customer loyalty. The location is a key to customer flow and has strong relationships with the quantity and amount of customer payment. Therefore I thought well about the reason and suppose that the original dataset includes random 2000 shops from thousands of thousands shops, which are distributed around the country, because there are thousands of thousands shops around the country, 2000 shops are relatively

negligible. It is no point to compare and link their location or geographic, for example, there is no meaning to analyze the locations of one restaurant at Swansea and one large grocery at central London. Few samples can not lead to a convincing conclusion, this is the risk of data mining that insufficient original dataset or lack of relative attributes reduce the accuracy of model.

6.3. Secondary pre-processing

Though I have apply the Pearson correlation with seven attributes, the category is string data type and could not be applied. Therefore I decide to transform the string data to numerical data.

Tab 6.2 Category data transformation

Data Transformation	
string	numerical
food	1
drink	1.1
leisure food	1.2
Baked pastries	1.3
fast food	1.4
snack	1.5
chinese food	1.6
hot pot	1.7
Soup / porridge / pot / casserole / stew	1.8
other	1.9
grocery	2
supermarket	2.1
Convenience store	2.2
leisure	3
internet café	3.1
medical health	4
pharmacy	4.1
Hairdressing / beauty / nail	5
beauty/Hairdressing	5.1

Tab 6.3 Samples of secondary preprocessing

category_level_1_name	category_level_1	category_level_2_name	category_level_2
food	1	drink	1.1
grocery	2	supermarket	2.1
food	1	drink	1.1
grocery	2	supermarket	2.1
food	1	leisure food	1.2
food	1	Baked pastries	1.3
food	1	fast food	1.4
food	1	leisure food	1.2
grocery	2	supermarket	2.1
grocery	2	supermarket	2.1
food	1	snack	1.5
food	1	Chinese food	1.6
food	1	fast food	1.4
food	1	fast food	1.4

6.4. Pearson Correlations

After pre-processing, I apply Pearson correlations again to explore the relationships among those attributes. The result are removed the repeated part of Tab 6.1.

The result is similar with the one without category. Only shop_id and location has no relationship with the category, customer average amount, score, quantity of comments, shop level have relationship with category. Especially total customer flow have weak positive relationship with category, including upper category and secondary category.

Tab 6.4 Pearson correlations of category

Correlations			
		category_level_1	category_level_2
	Pearson Correlation	.005	-.001
	Sig. (2-tailed)	.811	.958
	N	1999	1989
	Pearson Correlation	.002	-.004
	Sig. (2-tailed)	.913	.863
	N	1999	1989
	Pearson Correlation	.135**	.200**
	Sig. (2-tailed)	.000	.000
	N	1999	1989
	Pearson Correlation	.146**	.027
	Sig. (2-tailed)	.000	.266
	N	1708	1698
	Pearson Correlation	-.383**	-.322**
	Sig. (2-tailed)	.000	.000
	N	1708	1698
	Pearson Correlation	-.138**	-.106**
	Sig. (2-tailed)	.000	.000
	N	1999	1989
	N	1989	1989
	Pearson Correlation	.120**	.090**
	Sig. (2-tailed)	.000	.000
	N	1999	1989
*. Correlation is significant at the 0.05 level (2-tailed).			
**. Correlation is significant at the 0.01 level (2-tailed).			

6.5. Initial exploration for last two weeks

According to the goal of short-term prediction and previous analysis, the data may be periodic and have strong relationship with time, such as week or month; therefore I try to make descriptions of customer flow about the shop_988 (randomly selected) and total flow.

Figures 6.1 and 6.2 show periodicity. At Fig 6.1, there are a growing number of customer flow due more shops settled in koubei.com, and it appears a periodic wave. The wave is more obvious at Fig 6.2, separate shop represent a stable and periodic tendency. In conclusion, it is good to take use of time series.

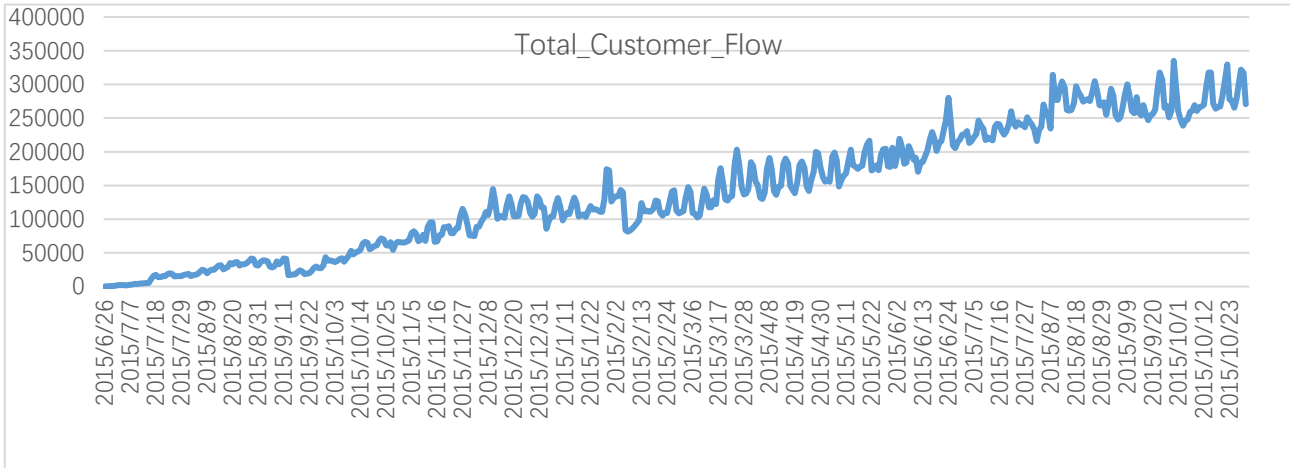


Fig 6.1 Total customer flow

According to the sales of shop_988, there are several futures worthy of attention. The first group of peaks is at the beginning of shop operation, usually some coupons voucher would be applied, therefore sales of the shop increased violently; the second peak is at 31 January, which is the Spring Festival like Christmas; the third group of peaks is at 4th -20th April, I have no idea with those peaks, which may be caused by some promotions. The last progressive and not much significant waves are from 5th Aug to 3rd Sep, that is due to the local Oktoberfest.

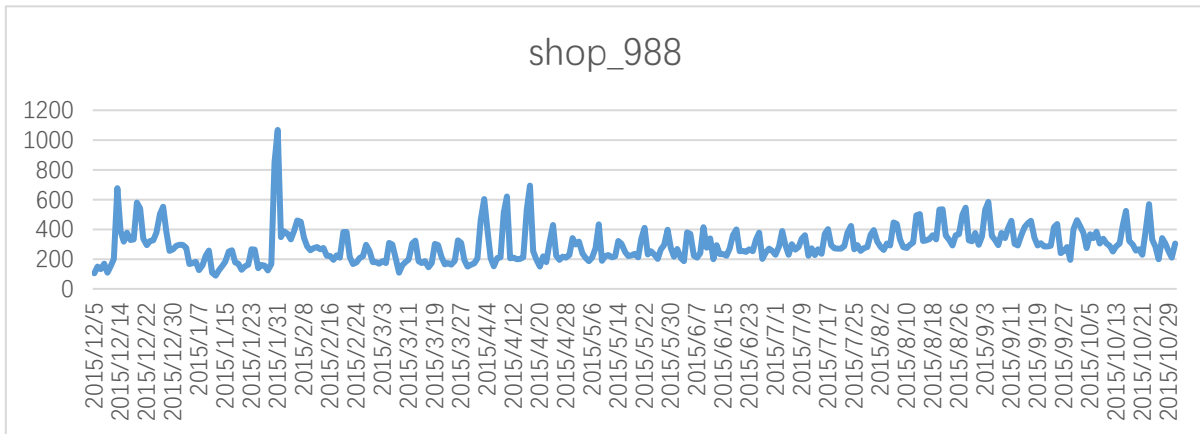


Fig 6.2 shop_988 customer flow

Customer Flows Forecasts For Shops On Koubei.com

Furtherly, I just apply the last two weeks to predict and simulate the customer flow. The I got the accuracy is 0.10694249, which is high, but it is a good beginning.

Tab 6.5 samples of last two weeks customer flow

shop _id	2015 /10/11	2015 /10/12	2015 /10/13	2015 /10/14	2015 /10/15	2015 /10/16	2015 /10/17
1	254	239	228	244	264	264	207
2	0	0	0	0	0	0	0
3	68	63	56	71	119	97	75
4	86	113	71	121	165	200	89
5	223	218	230	195	185	211	274
6	83	91	90	121	112	137	83
7	136	93	128	140	183	208	135
8	81	63	65	49	103	101	66
9	225	202	192	197	251	282	254
10	108	111	73	89	76	103	96

“ or ”, therefore I correct the double quotes with the function of find and replace. Then I start applying the time series to forecast.

```
@RELATION    dailysale
@ATTRIBUTE   saledate DATE "yyyy-MM-dd"
@ATTRIBUTE   shop1  numeric
@ATTRIBUTE   shop2  numeric
.....
@DATA
"2015-06-26",0,0,0,0,0,0,0,0,0.....
"2015-06-27",0,0,0,0,0,0,0,0,0.....
```

7.2. Linear Regression

=== Run information ===

Scheme:

```
LinearRegression -S 0 -R 1.0E-8 -num-decimal-places 4
```

Lagged and derived variable options:

-F

```
shop897,shop217,shop315,shop380,shop1350,shop1557,shop1745,shop1776,shop88,shop145 -L 1  
-M 7 -G saledate -dayofweek -weekend
```

Relation: random ten shops sales

Instances: 479

Attributes: 11

Transformed training data:

Instances were inserted in the training data for the following time-stamps (target values set by interpolation):

2015-12-12T00:00:00

The following training instances had missing values imputed via interpolation. Check source data as this may affect forecasting performance:

shop897:

Linear Regression Model

shop897 =

$$68.7421 * \text{DayOfWeek=tue,wed,thu,fri,sun,sat} + \\ -27.7134 * \text{DayOfWeek=wed,thu,fri,sun,sat} + \\ -26.1651 * \text{DayOfWeek=thu,fri,sun,sat} +$$

.....

=== Future predictions from end of training data ===

shop_id	897	217	315	380	1350	1557	1745	1776	88	145
18/10/2016*	342.6503	537.9685	177.9334	211.3795	473.9753	418.4723	392.1279	110.6071	504.4871	58.7622
19/10/2016*	317.9496	407.8618	132.0741	115.7626	348.9908	227.3192	296.8351	107.3004	352.0211	7.5146
20/10/2016*	215.1251	325.889	63.2099	54.7598	271.2557	222.9956	278.5379	46.7444	395.1403	-53.206
21/10/2016*	259.2167	347.6575	111.188	12.6191	398.3339	260.6086	286.917	62.4102	432.417	-39.3271
22/10/2016*	416.8838	259.0282	100.7445	62.2576	170.0608	264.506	344.4608	104.3583	387.7178	-52.9502
23/10/2016*	412.7362	236.7679	74.4788	56.6699	224.8721	297.6367	340.7884	118.726	483.7522	-35.4349
24/10/2016*	147.4886	103.0902	52.2061	-17.7567	231.1027	123.8881	162.3276	33.2966	357.2509	-43.102
25/10/2016*	121.3219	148.9129	113.3731	3.9057	300.8519	135.104	197.4757	38.8984	284.6133	-55.0346
26/10/2016*	230.4481	188.0697	107.077	39.2291	350.2382	193.2559	282.5566	88.73	258.1522	-28.6789
27/10/2016*	235.5953	279.6989	88.2727	89.7885	360.4496	166.9351	286.5795	70.8765	294.0873	-32.7068
28/10/2016*	367.5603	353.6998	147.2543	127.0627	441.8728	259.2593	309.3243	96.3836	349.4215	-6.7172
29/10/2016*	515.7296	399.0241	144.7883	216.6746	422.197	381.058	416.8163	176.9463	378.3202	26.0919
30/10/2016*	397.0032	348.1116	104.4842	199.7927	336.614	306.3406	356.1784	129.3056	396.5517	19.036
31/10/2016*	143.1082	280.0367	118.8677	120.2586	365.3529	217.437	223.7791	73.4247	292.3203	-13.406

Total number of instances: 480

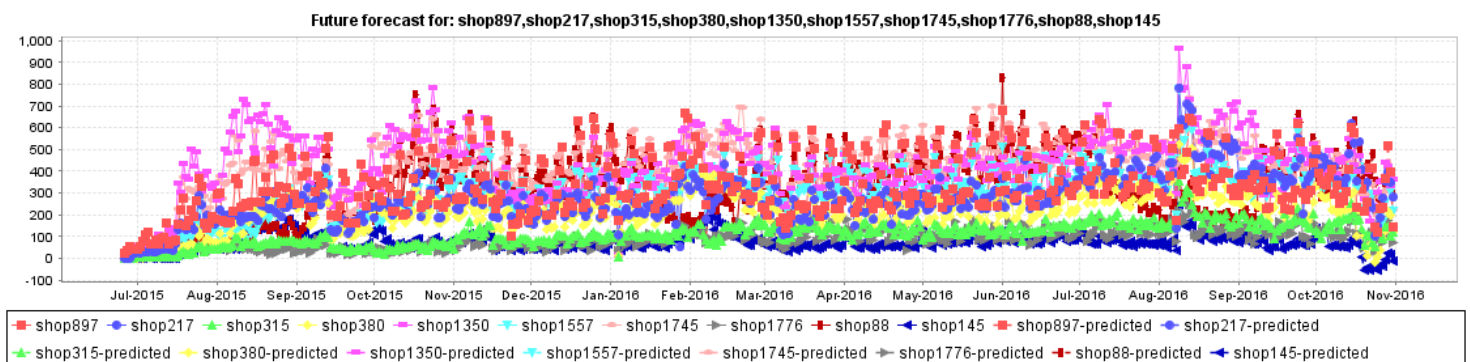


Fig 7.2 Sales and Predictions of Linear Regression for 10 shops

The first algorithm is linear Regression, I set 14 time units to forecast, and the result shows that total Root mean squared error is 162.3790, which is better than expected. At the end of this chapter, I will compare all the result.

7.3. Gaussian Processes

=== Run information ===

Scheme:

GaussianProcesses -L 1.0 -N 0 -K "PolyKernel -E 1.0 -C 250007" -S 1

.....

=== Future predictions from end of training data ===

shop_id	897	217	315	380	1350	1557	1745	1776	88	145
18/10/2016	266.5312	418.6031	188.7618	184.1074	507.6913	361.7787	417.014	106.9133	419.2785	63.9385
19/10/2016	314.0644	413.0813	176.7139	172.5219	474.4641	322.9419	414.1755	108.7278	447.1836	61.2094
20/10/2016	286.3172	395.2227	158.2986	165.0941	433.2143	309.4948	401.8172	97.6155	459.8483	45.2031
21/10/2016	360.6028	426.8901	167.4824	180.7044	465.1847	356.8042	417.8346	104.8545	501.799	46.4103
22/10/2016	529.4081	414.1821	163.7279	216.6581	415.3281	393.5152	474.9206	144.2578	505.3373	57.4379
23/10/2016	473.0584	373.5426	141.2246	208.48	323.7853	360.0983	458.1433	136.626	494.3283	50.3336
24/10/2016	240.882	309.9631	138.8485	118.5809	349.403	275.5103	348.3863	82.7884	424.6222	38.2158
25/10/2016	260.4122	338.1034	152.4443	136.3586	422.3021	304.0414	362.079	90.4016	435.3026	46.3254
26/10/2016	296.9624	356.7003	150.8998	155.5378	427.2397	312.7189	390.3863	97.2373	486.6375	48.4429
27/10/2016	310.3187	364.814	149.32	162.6052	422.1128	326.548	390.8494	99.8192	490.9219	46.7802
28/10/2016	391.9974	409.9507	165.4409	196.7698	466.6611	385.4886	427.1908	119.9766	531.8605	58.8753
29/10/2016	528.7672	425.9717	169.0061	241.8966	438.2236	418.6905	475.0522	155.3751	542.1449	71.0448
30/10/2016	495.9633	408.4015	151.6216	239.139	354.6104	383.7089	465.0818	143.1368	534.3899	67.8887
31/10/2016	287.9181	352.7424	150.8093	168.7447	406.8972	312.9317	376	97.0277	462.0212	56.6006

Total number of instances: 480

Future forecast for: shop897,shop217,shop315,shop380,shop1350,shop1557,shop1745,shop1776,shop88,shop145

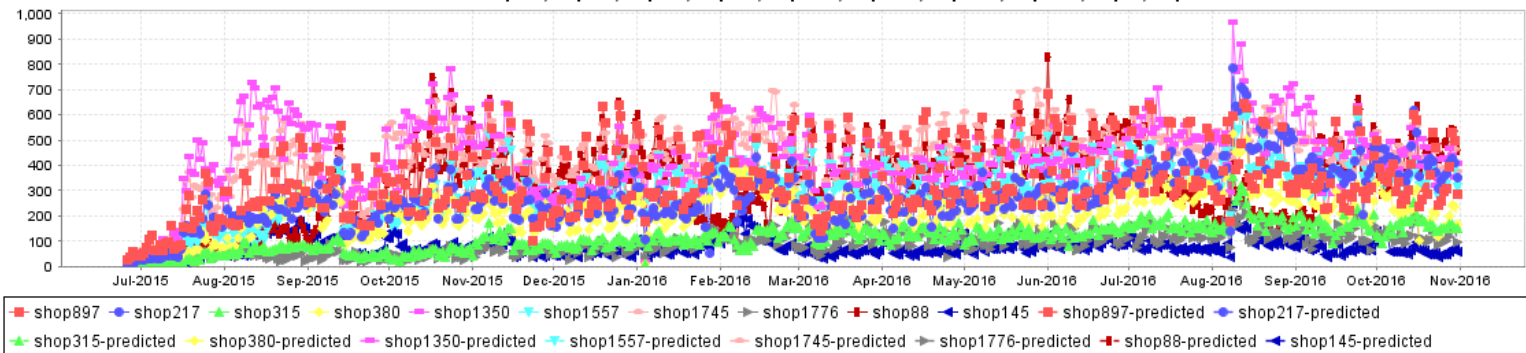


Fig 7.3 Sales and Predictions of Gaussian Processes for 10 shops

From EMSE, the total RMSE is 109.0213379, the result seems better than Linear Regression, besides, from the chart, we can see that the prediction of Gaussian Processes more stable, however the actual data is more sharp, overall, the tendency is consistent.

7.4. Multilayer Perceptron

==== Run information ====

Scheme:

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

==== Future predictions from end of training data ====

Total number of instances: 480

shop_id	897	217	315	380	1350	1557	1745	1776	88	145
18/10/2016	336.3357	584.3293	183.5354	157.2405	388.5548	391.1447	277.6356	96.4391	451.8238	82.6636
19/10/2016	477.4978	508.0507	126.6712	71.01	448.9702	312.8783	383.1386	134.9692	623.7801	82.6666
20/10/2016	462.8609	529.6046	146.6176	161.1042	476.319	351.4265	510.3739	165.697	708.9966	91.3347
21/10/2016	317.9336	626.0339	127.4909	126.1742	471.5874	250.9471	250.1796	80.1159	239.8162	85.3637
22/10/2016	412.9566	496.6289	142.7302	3.7601	267.9916	394.1233	110.6115	71.2984	-35.025	82.678
23/10/2016	614.5	653.583	129.0241	146.992	206.115	412.1077	386.4581	138.1721	817.296	82.6636
24/10/2016	365.6038	534.4089	104.7479	274.8634	476.22	107.5284	70.3722	129.1802	945.8394	82.6783
25/10/2016	-80.6501	218.925	245.3348	-97.6298	-163.513	51.9205	86.2185	4.3784	10.0945	82.6636
26/10/2016	130.6282	203.0937	335.7055	48.1406	99.5658	393.265	-300.762	356.2903	622.389	82.6661
27/10/2016	306.8591	487.76	182.2487	193.6405	-934.637	138.7524	-291.927	113.0004	38.2338	30447.38
28/10/2016	-149.597	-430.537	633.7401	375.0118	-2108.57	-11320.1	-404.338	372.0979	1296.257	87487.51
29/10/2016	-823.584	573.1076	634.0318	615.0568	-1774.04	-11320.1	-404.338	484.363	918.8018	87487.51
30/10/2016	-823.584	417.3699	634.0318	900.5957	-1548.1	-11320.1	800.1401	450.509	1053.895	87487.51
31/10/2016	-598.477	84.4731	244.9586	620.1557	-1548.1	-11320.1	-143.636	245.4407	1053.895	87487.51

Total number of instances: 480

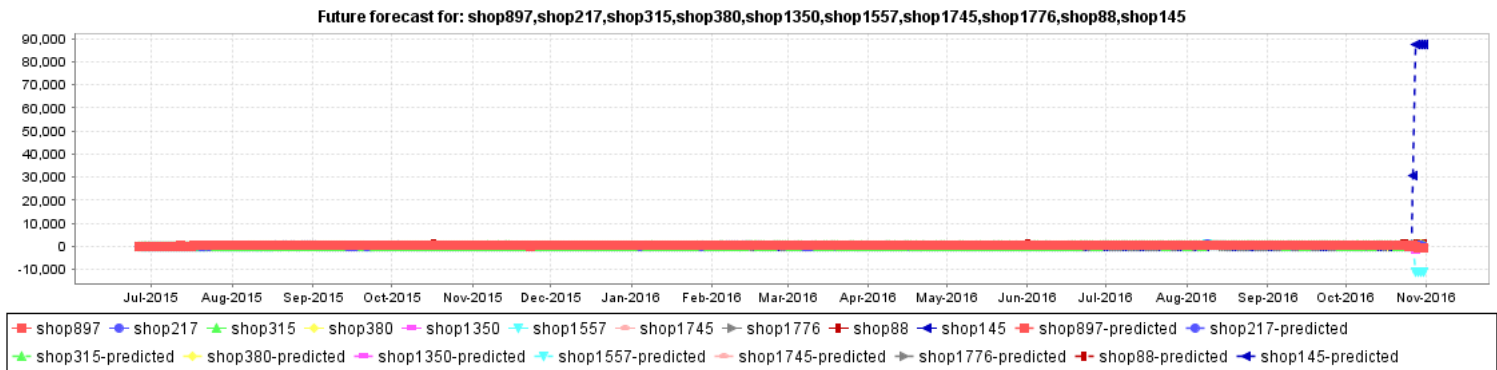


Fig 7.4 Sales and Predictions of Multilayer Perceptron for shop_888

The RMSE is 15138.1801, which is the worst in the results of three algorithms. But it is worth noting that the predictions of last week is closely matched. On the other hand, the predictions are negative number, I will take absolute value and transfer it into positive number.

7.5. SMOreg

==== Run information ====

=== Run information ===

Scheme:

SMOreg -C 1.0 -N 0 -I "RegSMOImproved -T 0.001 -V -P 1.0E-12 -L 0.001 -W 1" -K

"PolyKernel -E 1.0 -C 250007"

=== Future predictions from end of training data ===

shop_id	897	217	315	380	1350	1557	1745	1776	88	145
18/10/2016	266.8149	423.9511	164.404	222.9753	519.0863	377.3892	395.4064	89.9384	449.3393	59.3375
19/10/2016	328.5346	437.4215	181.3486	166.2873	453.0038	325.7109	388.7102	92.8916	370.8079	60.2272
20/10/2016	289.7453	442.6429	126.7172	193.2505	427.0328	344.8474	444.5008	91.9732	472.0193	37.0413
21/10/2016	425.0253	523.0387	188.9669	221.8328	492.4085	447.9527	457.0778	106.1856	515.5895	63.4102
22/10/2016	652.0673	545.6353	187.0823	311.7706	409.3677	483.2027	527.2142	187.9438	526.0132	77.1087
23/10/2016	561.6852	477.3389	147.1016	284.4444	310.1171	452.1037	458.3569	158.8581	453.1892	69.5686
24/10/2016	267.5375	367.4232	137.3931	189.0471	370.4327	286.4883	375.1368	77.0611	415.7458	50.39
25/10/2016	280.3923	384.8072	174.1286	175.1426	452.5012	296.9449	344.902	86.2406	350.4896	47.0064
26/10/2016	340.7421	412.2371	162.9965	181.5269	438.7651	297.5871	468.5922	104.7451	369.8164	46.853
27/10/2016	359.0834	444.974	150.2798	201.074	485.8674	343.8887	429.4538	93.2051	390.2114	51.8871
28/10/2016	489.3437	508.4356	202.2107	252.6556	519.714	385.4457	406.7917	123.7843	400.9353	66.2975
29/10/2016	569.816	504.0302	185.3475	292.7276	418.5941	443.167	449.4094	160.3566	361.9499	63.9818
30/10/2016	487.4022	507.5002	151.2484	314.9242	370.1673	386.1967	470.6513	139.5017	395.5713	62.2922
31/10/2016	265.4767	457.8172	161.7915	232.6015	470.652	298.4836	377.8701	94.7764	311.7393	46.0425

Total number of instances: 480

Future forecast for: shop897,shop217,shop315,shop380,shop1350,shop1557,shop1745,shop1776,shop88,shop145

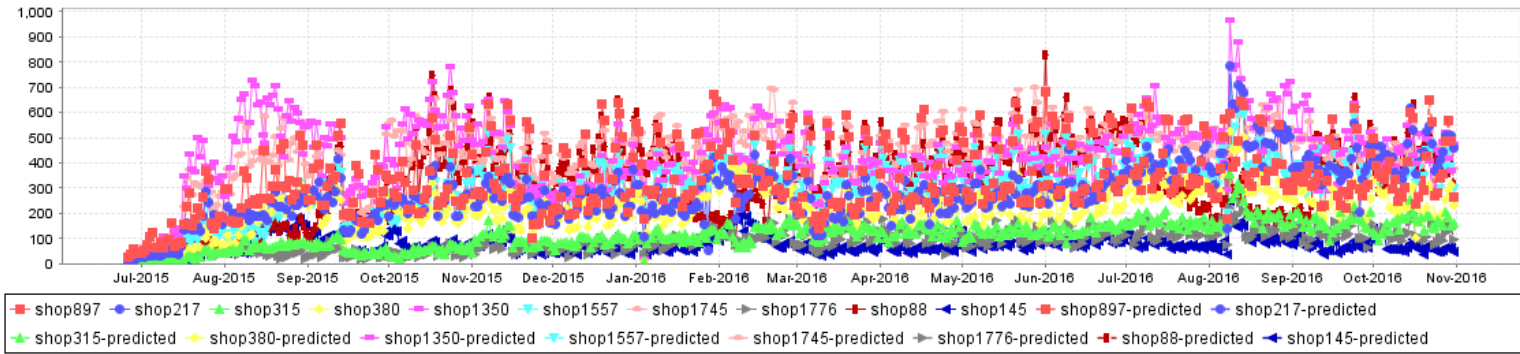


Fig 7.5 Sales and Predictions of SMOreg shop_888

RMSE is 111.9317088, which are similar with the ones of Linear Regression, but predictions of the two algorithms are greatly different.

7.6. Results comparison

In order to make a better choice, I make a table to compare predictions of four time series algorithms.

I collect all predications of four algorithms, and then compare them with actual data. By calculating RMSE, I can tell the accuracy of each algorithm. Obviously, the highest accuracy is from Gaussian Processes, the RMSE is 109.0213 and apparently lower than ones of other algorithms.

Therefore, the best model in time series is Gaussian Processes. In addition, I will take use of all the algorithms to ensemble.

Tab 7.1 Predictions comparison

Algorithm	RMSE
Multilayer Perceptron	15138.1801
Linear Regression	162.3790
Gaussian Processes	109.0213
SMOreg	111.9317

8. Neural Network

Before applying the neural network, I study some samples, and find that the dataset from koubei.com is a little unsuitable, because the neural network could deal with the dynamic data; unfortunately, the dynamic part of the dataset is daily sales, which means the rest shop information is static and excluded from the analysis model, like Tab 8.1. In order to meet the requirement, I intend to pre-process the data, and calculate the mean.

Tab 8.1 Sample of neural network data

shop_id	888	date	location	per_pay	score	comment_cnt	shop_level	category_level_1	category_level_2
888	7	10/12/2015	637	9	1	1	0	1	1.5
	10	11/12/2015							
	19	13/12/2015							
	12	14/12/2015							
	16	15/12/2015							
									

When I apply the neural network to Multilayer Perceptron neural network to predict the short-term customer flow, and the result shows those shop info are excluded, and only daily sales are left, because they are constant.

The warning:

The following independent variables are constant in the training sample and are excluded from the analysis: location, per_pay, score, comment_cnt, shop_level, category_level_1, category_level_2.

8.1. Secondary Pre-processing

At first I want to get the mean of daily sales, but there is a difficult, where the zero means the shop is closed at that day, therefore the day counts should not include those zero ones, so I use the excel formulate:

$$=\text{SUM}(\text{B2:B494})/\text{COUNTIFS}(\text{B2:B494}, ">0")$$

After applying the formulate, I got the mean data of daily sales and shop information:

Tab 8.2 Samples of mean and shops information

shop_id	shop_sale_avarage	location	per_pay	score	comment_cnt	shop_level	category_level_1	category_level_2
1	240	885	8	4	12	2	1	1.1
2	119	64	19			1	2	2.1
3	77	774	5	3	2	0	1	1.1
4	130	380	18			1	2	2.1
5	156	263	2	2	2	0	1	1.2
6	62	1139	13	3	1	0	1	1.3
								

8.2. Neural network model

Firstly, I plan to apply neural network to build a model to analyze the mean. Then I intend to combine mean and daily sales of time series to predict the short-term customer flow, and the mean will be temporary constant part at the model, actually, the mean may be calculated by sales of the last year.

I choose Multilayer Perceptron neural network, and the R tool is RSNNS, Stuttgart Neural Network Simulator (SNNS) is an excellent neural network simulation software developed by the University of Stuttgart, Germany, which is widely used by foreign neural network researchers. The content of the manual is extremely rich, while supporting the friendly Linux platform. And R and SNNS is connected to the RSNNS tool, in the R users can directly call the SNNS function command.

```
library(RSNNS)

data(shop_records.csv)

shop_records = shop_records [sample(1:nrow(shop_records),length(1:nrow(shop_record
s))),1:ncol(shop_records)]

shopValues= shop_records [,1:7]

shopTargets = decodeClassLabels(shopValues [,5])

shop_records = splitForTrainingAndTest(shopValues, shopTargets, ratio=0.15)

shop_records = normTrainingAndTestSet(shop_records)

model      =      mlp(shop_records$inputsTrain,      shop_records$targetsTrain,      size=5,
learnFunc="Quickprop",  learnFuncParams=c(0.1,  2.0,  0.0001,  0.1),maxit=100, inputsTest=
shop_records$inputsTest, targetsTest= shop_records$targetsTest)

predictions = predict(model, shop_records$inputsTest)

confusionMatrix(shop_records$targetsTest,predictions)
```

The result shows Relative Error is 23.6% according to Tab 8.4, which is a good figure and means neural network works on the dataset. Number of Hidden Layers is 1, and Number of Units in Hidden Layer is 7 based on the Tab 8.3, which includes all attributes, I am little surprised that location could take the most importance according to Tab 8.5, because I thought the location may lose effect due to so little data compared to the whole shops at database. From the result, I think though the correlation between location and total sales is weak even extremely weak, but the location has an important influence on the mean of sales.

Tab 8.3 Network Information

Input Layer	Factors	1	shop_level
		2	location
		3	per_pay
		4	score
		5	comment_cnt
		6	category_level_1
		7	category_level_2
		Number of Units ^a	
	Number of Hidden Layers		1
	Number of Units in Hidden Layer 1 ^a		7
	Activation Function		Hyperbolic tangent
	Dependent Variables	1	shop_sale_avarage
	Number of Units		1
	Rescaling Method for Scale Dependents		Standardized
	Activation Function		Identity
	Error Function		Sum of Squares
a. Excluding the bias unit			

Tab 8.4 Model information

Trainin g	Sum of Squares Error	99.712
	Relative Error	.168
	Stopping Rule Used	Maximum number of epochs (100) exceeded
	Training Time	0:00:05.89
Holdout	Relative Error	.236
Dependent Variable: shop_sale_avarage		

Tab 8.5 Independent Variable Importance

	Importance	Normalized Importance
shop_level	.016	2.9%
location	.541	100.0%
per_pay	.102	18.8%
score	.009	1.8%
comment_cnt	.173	31.9%
category_level_1	.123	22.8%
category_level_2	.036	6.6%

At this model, I apply Hyperbolic tangent as activation function, and Number of Hidden Layers is 1. The result shows that Sum of Squares Error is 99.712, and each variable is normalized, get the normalized importance.

However, I could not get a prediction from this code and the result, so I decide to apply back propagation neural network.

8.3. Back Propagation Neural Network

When I choose the Neural Network, with the consideration of time cost, advantages and disadvantages of different algorithms, I decide to apply Back Propagation Neural Network, which is a mature one. BP neural network in terms of network theory or performance has been relatively mature. Its outstanding advantage is that it has a strong nonlinear mapping ability and flexible network structure. The number of layers in the network, the number of neurons in each layer can be arbitrarily set according to the specific circumstances, and with the structural differences in its performance is different.

In the case, I apply math API to take use of BPNN. math is a high-level neural network API. math is written by pure Python and based on variable and . math is able to quickly convert your ideas into results for support of quick experiments. Another reason why I choose math is that it is simpler than theano and tensorflow, which I tried to apply but failed due to complex parameters and Linux-preferrable operation.

According to BP Neural Network, Increasing the number of hidden layers can reduce the network error and improve the accuracy, but it complicates the network, thus increasing the training time of the network and the tendency of "over-fitting". Generally speaking, Neural Networks should be designed to give priority to the three-tier network (that is, one hidden layer). In general, increasing the number of hidden layer nodes to obtain a lower error has the better training effect and easier way to achieve than the increasing number of hidden layer.

In order to insurance generalization of the model, I re-select 10 shops randomly, then import the daily sales, shop ids are 897, 217, 315, 380, 1350, 1557, 1745, 1776, 88, and 145.

The main code I take use of BPNN as following, and the complete code is attached as Appendix A:

```
-----
HID_NODE_SIZE = 10  # Number of hidden nodes
MAX_TRAIN = 1500  # Maximum number of training sessions
LEARN_RATE = 0.3  # Learning rate
MOMENTUM = 0.1  # momentum factor
# Generates random numbers within interval [a, b)
# Generate the matrix of size I * J, the default zero matrix (of course, also use NumPy to speed up)
# Function sigmoid, where tanh is used
# Function sigmoid derived function, in order to get the output (ie: y)
    # Activate all nodes of the neural network
    # Build weight (matrix)
    # Set to random value
    # Finally, the momentum factor (matrix)
    # Activate the input layer
    # Activate the hidden layer
    for j in range(self.nh):
        # Activate the output layer
    for k in range(self.no):
        # Calculate the output layer error
        # Calculate the error of the hidden layer
        # Update the output layer weight
        for j in range(self.nh):
            # Update the input layer weights
# Calculation error
error = 0.0
    # Standardized test results
    # Anti-standardization
# print(' Input layer weight:')
    # print(self.wi[i])
# print()
```

```

# print(' Output layer weight:')

# N: learning rate

# M: momentum factor

# csv file

# Read the training files and organize the data into the training set format [[[input vector], [output
vector]], [[input vector], [output vector]], ..., [[input vector], [ Output vector]]]

# Split the entire data set into training sets and test sets

train_data_list = []

# print(len(result),len(result[0]),len(result[0][0]),len(result[0][1]))

# Create a neural network: the input layer has ipt nodes, the hidden layer has hdn nodes, the

# Look at the weight of training (of course, you can consider the weight of the training to a long
time

# n.weights()

if __name__ == '__main__':

    main()

```

Then I import 2000 shops daily sales in order to predict the short-term 14 days sales, the results are different due to different parameters. At the beginning of the python code, I set four static variables, including number of hidden nodes, maximum number of training sessions, learning rate, momentum factor, which have effect on the performance of the model. Therefore, I test different parameters to try to find the best one.

(the particular number is just a example)

```
HID_NODE_SIZE = 10 # Number of hidden nodes
```

```
MAX_TRAIN = 1500 # Maximum number of training sessions
```

```
LEARN_RATE = 0.3 # Learning rate
```

```
MOMENTUM = 0.1 # momentum factor
```

8.3.1. Number of hidden layer

At first, I adjust number of hidden layer, I set nine different values as following Tab 8.6. Then running the code, I get nine prediction sets of 14 days sales, therefore I calculate RMSE between the predictions and the real data. In fact, 4 hidden layer results in lowest RMSE, which is the best one. In addition, more hidden layers are not necessarily the best one according to Fig 8.1, sometimes more hidden layers not only cost more time and energy, but also lead to worse result.

Tab 8.6 results of different number of hidden layer

number of hidden layer	RMSE
2	109.0009
3	106.4719
4	104.3432
5	108.8294
6	110.1857
7	109.6772
8	118.6622
10	127.9064
15	146.7473

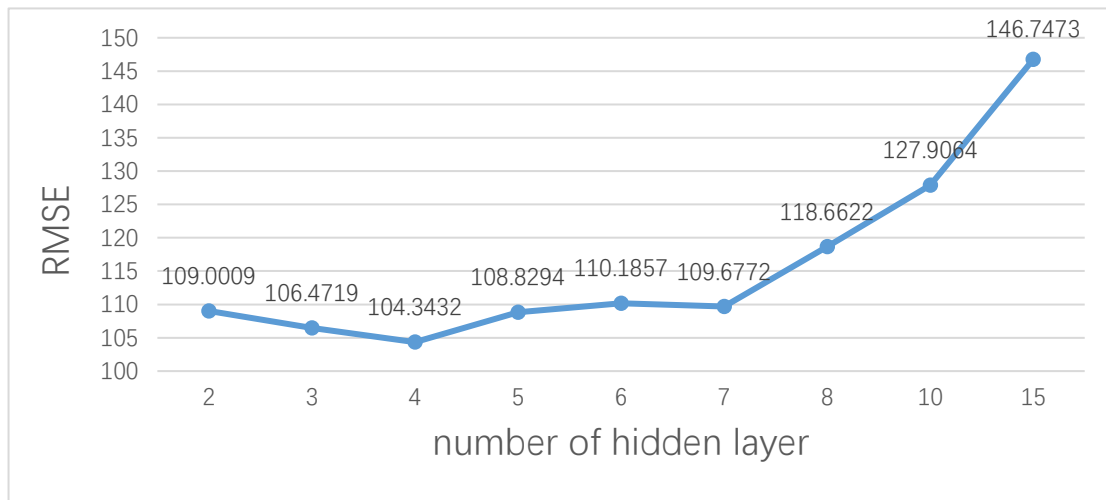


Fig 8.1 results of different number of hidden layer

8.3.2. Maximum number of training

Another important variable is maximum number of training. More trained the model is, more mature it is. However, it is not perfect to take the as more number of training session as possible in reality, because it would cost more time, money and energy. In this case, I set 8 different values of maximum number of training, then 8 sets values of prediction are made. According to Tab 8.7, RMSE is decreasing violently from number of training as 1200, besides, RMSE is very close between 1200 and 3000. Therefore under the premise of accuracy assurance, I think the maximum number of training could be applied as 1500.

Tab 8.7 results of different of maximum number of training

maximum number of training	RMSE
500	110.9241
1000	109.7937
1200	108.9658
1500	108.8294
2000	108.9518
2500	108.8017
3000	108.8318
4000	109.7755

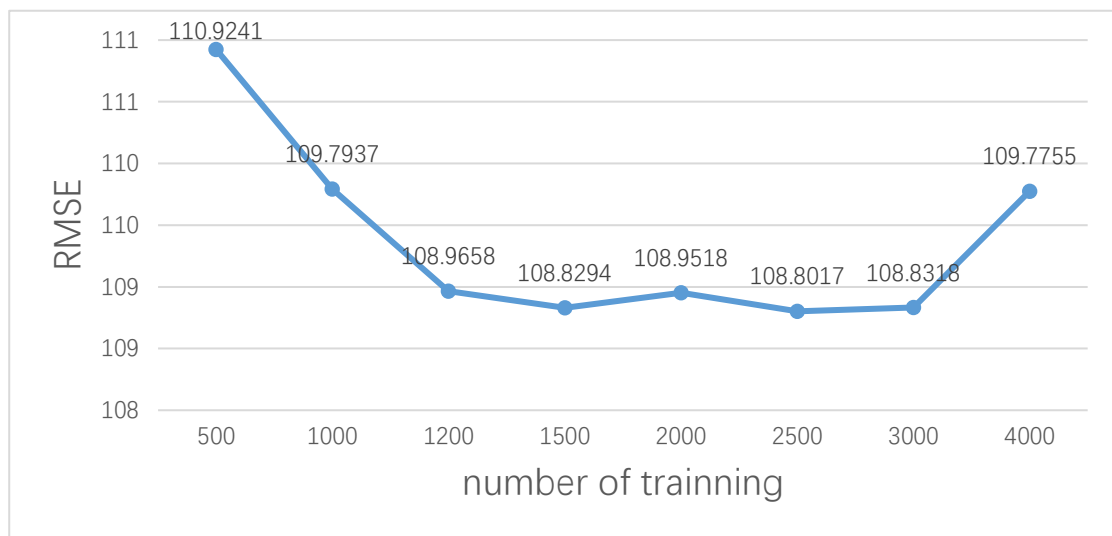


Fig 8.2 results of different of maximum number of training

8.3.3. Result of prediction of BPNN

After different contrast trainings, I set number of hidden layer as 4, maximum number of training as 1500, then the result is as following:

Tab 8.8 Samples of prediction results

Shop_ID/date	18/10/2016	19/10/2016	20/10/2016	
897	440.3379197	454.4481134	455.1283327	
217	531.3113222	540.7916007	524.5468494	
315	229.5103541	236.4444987	220.1273606	
380	176.6115477	222.8277877	248.6282285	
1350	515.691454	523.1119159	521.3824584	

In order to compare the performance of five algorithms, I run the code of time series algorithms,

then predict the 14 days sales of random 10 shops, the specific predictions could be referred to Appendix B.

Tab 8.9 RMSE of five algorithms of 10 random shops

algorithm	RMSE
Multilayer Perceptron	15138.1801
Linear Regression	162.3790
Gaussian Processes	109.0213
SMOreg	111.9317
BPNN	104.3432

The result shows the 14days sales of each shop from 18/10/2016 to 31/10/2016, and the RMSE of 10 shops is 104.3432, which is better than any time series algorithms, MultilayerPerceptron and SMOreg, LinearRegression, GaussianProcesses.

9. Ensembling

At this chapter, I try to ensemble the different models and result in better accuracy; In this case, I mainly take use of five algorithms, including Multilayer Perceptron, Linear Regression, Gaussian Processes, SMOReg, BPNN. Therefore, I intend to merge the results of five algorithms by weighted average and simple average.

9.1. Weighted average

Weighted average means different portions of prediction values of different algorithms, at this case, RMSE is the accuracy of model, and therefore I take advantage of RMSE to calculate the portion of each algorithm.

At first, because RMSE of Multilayer Perceptron is very much larger compared to ones of other algorithms, I set up two groups, one is calculating prediction of Multilayer Perceptron, and the other is calculating data not including the prediction of Multilayer Perceptron. Those two groups are used for comparison.

According to RMSE, I calculate the proportion of each algorithm like 0.9688, as we know, the less RMSE is, the more accurate model is, then calculate 1 minus the proportion and get the weight. In comparison, at the second group, I ignore the RMSE of Multilayer Perceptron, and then obtain the weight.

Next, I calculate the prediction of different weighted average. I attach the specific data as Appendix C.

RMSE of Weighted average (not including Multilayer Perceptron) is 96.2661, which is even better than the RMSE of BPNN, as we could see; BPNN is the best one in those algorithms. It demonstrates ensembling learning is an efficient and useful tool.

Tab 9.1 Weighted average

algorithms	RMSE	Weighted average(including Multilayer Perceptron)		Weighted average(not including Multilayer Perceptron)	
		proportion	weight	proportion	weight
Multilayer Perceptron	15138.1801	0.9688	0.0078	--	--
Linear Regression	162.3790	0.0104	0.2474	0.3330	0.2223
Gaussian Processes	109.0213	0.0070	0.2483	0.2236	0.2588
SMOReg	111.9317	0.0072	0.2482	0.2295	0.2568

Customer Flows Forecasts For Shops On Koubei.com

BPNN	104.3432	0.0067	0.2483	0.2140	0.2620
RMSE		151.9182		96.2661	

Tab 9.2 values of Weighted average (not including Multilayer Perceptron)

Date/shopid	897	217	315	380	1350	1557	1745	1776	88	145
18/10/2016	329	476	191	198	505	384	430	120	477	66
19/10/2016	355	452	184	171	454	317	399	122	428	55
20/10/2016	316	426	145	170	419	318	410	106	479	30
21/10/2016	391	467	178	178	471	367	424	116	511	39
22/10/2016	547	457	174	220	375	389	478	158	511	46
23/10/2016	503	420	142	211	340	379	455	152	504	46
24/10/2016	273	338	128	137	363	272	362	87	444	32
25/10/2016	279	352	153	148	419	281	359	92	408	30
26/10/2016	334	373	149	163	428	297	421	110	419	37
27/10/2016	347	396	142	183	439	309	413	101	443	36
28/10/2016	435	450	172	215	479	359	423	122	474	50
29/10/2016	541	474	168	258	448	410	477	158	476	60
30/10/2016	487	460	156	260	533	380	468	141	485	56
31/10/2016	293	398	162	200	581	313	393	109	412	42

9.2. Simple average

In this case, I also set two groups, the one is calculating values of Multilayer Perceptron, the other one is not calculating values of Multilayer Perceptron. Therefore, the two groups have different weight, separately 0.2 and 0.25.

According to Tab 9.2, RMSE of the Simple average (not including Multilayer Perceptron) is much better than one of Simple average(including Multilayer Perceptron), RMSE of the Simple average (not including Multilayer Perceptron) is 97.6399, which is close to one of Weighted average(not including Multilayer Perceptron).

Tab 9.3 Simple average

algorithms	RMSE	weight(including Multilayer Perceptron)	weight((not including Multilayer Perceptron)	RMSE
Multilayer Perceptron	15138.1801	0.2	0.25	
Linear Regression	162.3790	0.2	0.25	
Gaussian Processes	109.0213	0.2	0	
SMOreg	111.9317	0.2	0.25	
BPNN	104.3432	0.2	0.25	

Customer Flows Forecasts For Shops On Koubei.com

Tab 9.4 values of simple average (not including Multilayer Perceptron)

Date/shopid	897	217	315	380	1350	1557	1745	1776	88	145
18/10/2016	329	478	190	199	504	385	429	120	478	66
19/10/2016	354	450	182	169	450	314	395	121	425	53
20/10/2016	312	422	142	165	413	315	405	104	476	27
21/10/2016	386	463	175	172	468	363	418	114	508	37
22/10/2016	542	450	171	214	368	384	473	156	506	42
23/10/2016	499	414	139	206	336	376	450	151	503	43
24/10/2016	268	329	125	131	358	267	355	85	440	29
25/10/2016	273	344	152	143	414	276	353	90	403	27
26/10/2016	330	366	148	159	425	293	416	109	413	34
27/10/2016	343	391	140	179	437	304	408	100	437	34
28/10/2016	433	447	171	212	478	355	419	121	469	48
29/10/2016	540	471	167	256	446	410	475	159	472	59
30/10/2016	483	456	154	258	524	377	463	141	482	55
31/10/2016	287	394	160	197	571	309	387	108	407	40

10. Conclusion

In this case, I take analysis of daily sales of 2000 shops on Koubei.com, based on CRISP-DM methodology. In particular, I take use of variable tools including EXCEL, TABLEAU, WEKA, PYTHON, and take the six steps of business understanding, data understanding, data preparation, modeling, evaluation, deployment. The two main analysis algorithms are Time Series and Neural Network, then ensembling learning is applied. There are several conclusion as following:

10.1. Time series prediction

WEKA is mainly used to apply Time Series. According to the characteristics of shop sales and other info, there are four algorithms considered, namely Linear Regression, Gaussian Processes, Multiplayer Perceptron, SMOreg. Different algorithms have different applications, therefore the results, namely MSE and predictions, are variable.

Though at chapter 7, I just choose one shop to run four algorithms, and get the results, In order to avoid the particular characteristics of one shop, I choose 10 shops randomly, and keep the real data from 18/10/2016 to 31/10/2016, and refer to the previous data to predict 14 days sales. The values of predictions are attached as Appendix B.

The results of RMSE shows Gaussian Processes is the best one of four, because RMSE is 109.0213, much better than the other of three algorithms, the RMSE of SMOreg is approaching to the one of Gaussian Processes, the difference between Gaussian Processes and SMOreg is just 2.9104. The performance of Linear Regression is not as good as the previous two algorithms, which is 162.3790. By contrast, RMSE of Multilayer Perceptron is beyond my expectation, which is 15138.1801, is nearly 15 times than one of Gaussian Processes.

Therefore, the performance of Gaussian Processes is best among the four algorithms, in terms of single algorithm.

Tab 10.1 Predictions comparison

algorithms	RMSE
Multilayer Perceptron	15138.1801
Linear Regression	162.3790
Gaussian Processes	109.0213
SMOreg	111.9317

10.2. Neural network

In consideration of maturity of algorithm and cost, I apply Back Propagation Neural Network to predict the sales of each shop. According to Back Propagation Neural Network, there are four variables, two important factors are number of layers in the network, maximum of training sessions. Therefore, I adjust the number of hidden layer and maximum number of training in this case, in order to find the least RMSE.

At first, I adjust the number of hidden layer from 2 to 15, so I have 9 sets of predictions, then I calculate RMSE of each set. The result shows the best number of hidden layer is 4. But from the chart, it could be concluded that RMSE of hidden layer from 2 to 7 is little difference, all of them vary from 104 to 110, therefore, 3 hidden layers could be considered due to cost and time in reality. In the case, I decide to apply 4 hidden layers.

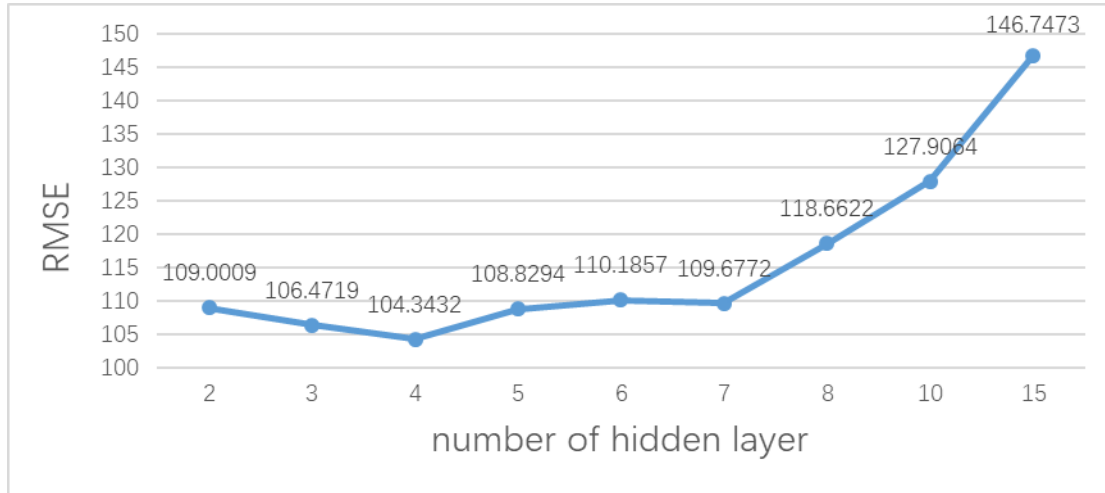


Fig 10.1 results of different number of hidden layer

Next, I change the maximum number of training from 500 to 4000. From the chart, it is easy to find the lowest RMSE, which is 1500 as maximum number of training. Furtherly, the graphic is U-shaped and parabola, RMSE of 1200, 1500, 2000, 2500, 3000 are little difference, starting from 2000, RMSE drops sharply. In reality, 1200 may be a better choice in term of cost and time. In this case, I choose 1500 as maximum number of training, which RMSE is 108.8294.

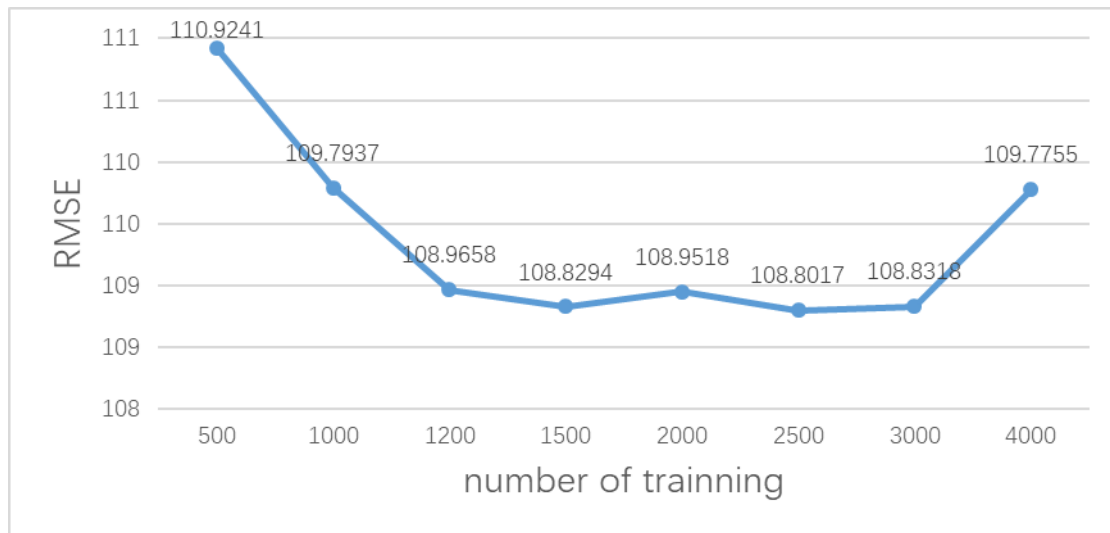


Fig 10.2 results of different of maximum number of training

10.3. Ensembling

After calculating the results of Time series and Neural Network, ensembling learning is applied

and expected to have a better results. In this case, I adopt two simple ensembling learning, one is weighted average, and the other is simple average.

At first, because RMSE of Multilayer Perceptron is very much larger compared to ones of other algorithms, I set up two groups, one is calculating prediction of Multilayer Perceptron, and the other is calculating data not including the prediction of Multilayer Perceptron. According to RMSE, I calculate the proportion of each algorithm, as we know, the less RMSE is, the more accurate model is. In comparison, at the second group, I ignore the RMSE of Multilayer Perceptron, and then obtain the weight. (the specifical data attached as Appendix C.)

RMSE of Weighted average (not including Multilayer Perceptron) is 96.2661, which is even better than any single algorithm. It demonstrates ensembling learning is an efficient and useful tool.

Tab 10.2 Weighted average

algorithms	RMSE	Weighted average(including Multilayer Perceptron)		Weighted average(not including Multilayer Perceptron)	
		proportion	weight	proportion	weight
Multilayer Perceptron	15138.1801	0.9688	0.0078	--	--
Linear Regression	162.3790	0.0104	0.2474	0.3330	0.2223
Gaussian Processes	109.0213	0.0070	0.2483	0.2236	0.2588
SMOreg	111.9317	0.0072	0.2482	0.2295	0.2568
BPNN	104.3432	0.0067	0.2483	0.2140	0.2620
RMSE	--	151.9182		96.2661	

On the other hand, I apply simple average to ensemble the prediction values. Corresponding, I set two group, one including Multilayer Perceptron, the other one not including Multilayer Perceptron. The result shows the performance of group without Multilayer Perceptron is much better than the other one. In addition, RMSE of simple average is 96.2661, which is little difference with one of weighted average. In reality, if the organization does not require high accuracy of model, simple average may be applied to save cost at this case.

11. Reflection

11.1. What I learn

At this case, I learn much professional and logistical knowledge. Data mining course lays a solid foundation for me. During the course, I learned the CRISP-DM, and got the basic and standard flow of data mining, therefore I could carry on one complete experiment, namely data of Koubei.com. Then I studied WEKA and TABLEAU, those two analysis tools are very useful and important for my next work. Besides, I also learned TABLEAU at its official website.

Then I finished four algorithms of Time series, I learned difference and characteristics of four algorithms, and applied all of them to predict short-term values, RMSE of each algorithm is different due to variable characteristics.

The most difficult part of analysis is neural network. Firstly, I learn it at Youtube, and find R and Python could have the package of Neural Network, after comparing them, I choose Python due to generalization and maturity. Then I study the 'math' package, and BP Neural Network, and find two parameters could be adjusted to product better results. Finally, I successful apply BP Neural Network to predict the sales of 10 shops.

Another important knowledge is ensembling learning. Before I take use of ensembling learning, I used to applying single model, but it is very useful after I apply it to my case at the first time. In this case, I just learn weighted average and simple average, whose performances are better than one of single algorithm.

11.2. Difficulties

What makes me annoying and feel difficult is repeating pre-processing. In order to meet the requirements of different algorithms, the format, attributes and number of data have to be adjusted and changed. At this case, I process data format four times.

Another difficult is Neural Network. Though I learned the basic knowledge of NN, when I am faced with much data and code, I really feel no way to start. At first, I hesitate to choose R or Python, after learning some online courses at Youtube, I decide to apply Python. Then it takes me a few days to learn the Python package and code. Fortunately, I mastered the knowledge of BPNN.

11.3. Expectations

There are several regrets and expectations about the case. I think the result would be better if the several expectations could be met.

(1) Data used to analyze

In fact, I just take use of daily sales, the shop info is not used. At first, I import shop attributes into Neural Network, but I find there is no difference between the result with shop info and the one without shop info, maybe the shop attributes are static variables. Therefore, I delete the shop attributes. I think shop attributes are important factors, which may be influential to some aspects. In

the future, some algorithms could be tested and validate the expectations.

(2) The minimal amount of data

In this case, I apply all data provided from Koubei.com, from June 2015 to October 2016. However, I think so much data may be not necessary. It is important that find out the minimal amount data, perhaps sales data of three months is enough, maybe just two months. In the future, I intend to find the minimal amount data if I carry on the project again.

(3) Ensembling learning

In this case, I just apply two simple ensembling learning methods. Facts have proved that results after ensembling learning are more efficient than ones of single algorithm. What a pity is that I have no enough time to learn more methods of ensembling learning. If I have enough time in the future, I will apply more ways of ensembling learning.

Reference

- 1 Alibabagroup.com. (2017). 阿里巴巴集团. [online] Available at: <http://www.alibabagroup.com/cn/global/home> [Accessed 7 Jun. 2017].
- 2 Antgroup.com. (2017). 蚂蚁金服官网-为小微企业和消费者提供普惠金融服务. [online] Available at: <https://www.antgroup.com/> [Accessed 7 Jun. 2017].
- 3 Koubei.com. (2017). 口碑. [online] Available at: <https://www.koubei.com/> [Accessed 7 Jun. 2017].
- 4 Wirth, Rüdiger, and Jochen Hipp. "CRISP-DM: Towards a standard process model for data mining." *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*. 2000.
- 5 M. Johnston, G. Marshall, "Introduction to sales management in the Twenty-First century," in *Sales force management*. New York: Routledge, 2013:2-3.
- 6 Paul H. Selden (December 1998). "Sales Process Engineering: An Emerging Quality Application". *Quality Progress*: 59–63.
- 7 Bain & Company. (2015, Nov 23). Management Tools - Customer Relationship Management [Online]. Available: www.bain.com. Retrieved.
- 8 D. Lee, JR. Watterkemper, RW. Seára, "Redefining CRM," *CRM Magazine*, Jul 2000.
- 9 J. Anton, "The past, present and future of customer access centers," *International Journal of Service Industry Management*, vol.11, pp. 120-30, May 2000.
- 10 E. Chablo, "The importance of marketing data intelligence in delivering successful CRM," *Customer Relationship Management*, pp. 57-70, 2000.
- 11 C. Imhoff, L. Loftis, J.G. Geiger, *Building the customer-centric enterprise: Data warehousing techniques for supporting customer relationship management*. New York: Wiley, 2001.
- 12 Van Rhodes. (Mar 2002). *SAS Solutions for Customer Relationship Management* [Online]. Available: <http://www.sas.com/solutions/crm/dmreview.pdf>
- 13 Gartner Group. (2017). [Online]. Available: <http://www.gartner.com>.
- 14 C. Gurău, A.Ranchhod, R.Hackney, "Customer-centric strategic planning: Integrating CRM in online business systems". *Information technology and management*, vol.4, pp. 199-214, Apr 2003.
- 15 Wal-Mart. (2017). [Online]. Available: <https://www.walmart.com/>
- 16 KDnuggets. (1989, August 20). KDD-89: IJCAI-89 Workshop on Knowledge Discovery in Databases. [Online]. Available: <http://www.kdnuggets.com/meetings/kdd89/index.html>
- 17 Advancement of Artificial Intelligence. (1995). The First International Conference On Knowledge Discovery And Data Mining. [Online]. Available: <http://aaai.org/Conferences/KDD/kdd95.php>
- 18 Data Mining and Knowledge Discovery. (2017). [Online]. Available: <https://link.springer.com/journal/10618>
- 19 Artificial Intelligence Review. (2017). [Online]. Available: <http://www.springer.com/computer/ai/journal/10462>
- 20 IBM. (2017). [Online]. Available: www.ibm.com.
- 21 SAS. (2017). [Online]. Available: www.sas.com.

- 22 Oracle. (2017). [Online]. Available: www.oracle.com.
- 23 Wayne Duggan. (2015, Aug, 17). Future of E-Commerce? [Online]. Available: <https://finance.yahoo.com/news/does-o2o-mean-future-e-233503739.html>.
- 24 R. Groth, "Examples for each study performed are provided, with tips on how these can be performed on corporate database systems," 2nd edition, *Data mining: building competitive advantage*. PTR: Prentice Hall, 2000, pp: 288-310.
- 25E. Keogh, J. Lin, S. H. Lee, H. V. Herle, "efficiently finding the most unusual time series subsequence", *Proceedings of the 5th IEEE International Conference on Data Mining*, vol. 11, no.1, pp.1-27,2007.
- 26 Q. Yang, X. Wu, "10 challenging problems in data mining research," *International Journal of Information Technology & Decision Making*, vol.5, no.4, pp. 597-604, 2006.
- 27 R. Agrawal, C. Faloutsos, A. Swami, "Efficient similarity search in sequence databases," *Foundations of data organization and algorithms*, pp.69-84, 1993.
- 28 Eamonn Keogh. (2017). [Online]. Available: <http://www.cs.ucr.edu/~eamonn/>
- 29
- 30 C. Apte, F. Damerau, S. Weiss, "Text mining with decision rules and decision trees," *Proceedings of the Conference on Automated Learning and Discovery*, Pittsburgh, USA, 1998, pp.62 - 68.
- 31 J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol.1, no.1, pp.81-106,1986.
- 32 C. Orlaru, L. Wehenkel, "A complete fuzzy decision tree technique," *Fuzzy sets and systems*, vol.138, no.2, pp.221-254, 2003.
- 33 Wikipedia. (2017). *Sigmund Freud* [Online]. Available: https://en.wikipedia.org/wiki/Sigmund_Freud
- 34 W. S. McCulloch, W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of mathematical biology*, vol.5, no.4, pp. 115-133, 1943.
- 35 D. O. Hebb, *The organization of behavior; a neuropsychological theory*. Oxford, England: Wiley, 1949.
- 36 F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological review*, vol.65, no.6, pp.386, 1958.
- 37 B. Widrow, M. E. Hoff, "Associative storage and retrieval of digital information in networks of adaptive 'neurons'", *Biological Prototypes and Synthetic Systems*. New York, USA: Springer, 1962, pp. 160-160.
- 38 L. Xu, "Bayesian Ying–Yang machine, clustering and number of clusters," *Pattern Recognition Letters*, vol.18, no.11, pp.1167-1178, 1997.
- 39 Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.
- 40 T.G. Dietterich, "Ensemble Methods in Machine Learning," In *Multiple Classifier Systems*, Berlin, Heidelberg, 2000, vol 1857, pp.1-15.
- 41 T. G. Dietterich, "Ensemble learning," In *The handbook of brain theory and neural networks*, Cambridge, MA: MIT Press, 2002, pp.110-125.
- 42 Z. H. Zhou, *Ensemble methods: foundations and algorithms*. Cambridge, UK: CRC press, 2012.

- 43 L. K.Hansen, P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol.12, no.10, pp. 993-1001,1990.
- 44 K. J. Cherkauer, "Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks," In *Working notes of the AAAI workshop on integrating multiple learned models*, Madison, 1996, pp. 15-21.
- 45 F. J. Huang, Z. Zhou, H. J. Zhang, et al. "Pose invariant face recognition". In *Automatic Face and Gesture Recognition Fourth IEEE International Conference*, Grenoble, France, 2000, pp. 245-250.
- 46 D. Silver, A. Huang, C. J. Maddison, et al, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol.529, no.7587, pp. 484-489, 2016.
- 47 Wirth R, Hipp J. CRISP-DM: Towards a standard process model for data mining[C]// European Planetary Science Congress 2008. European Planetary Science Congress 2008, 2008.
- 48 Cios K J, Kurgan L A. Trends in Data Mining and Knowledge Discovery[J]. 2005:200--2.
- 49 Tableau. (2017). [Online]. Available: <https://www.tableau.com/>
- 50 M. J. Sirgy, A. S. Coskun, "A path analytic model of store loyalty involving self-concept, store image, geographic loyalty, and socioeconomic status," *Journal of the Academy of Marketing Science*, 1985, vol.13, no.3, pp.265-291.

Appendix A: Neural Network code

```
import math
import random
import string
import csv

HID_NODE_SIZE = 10 # Number of hidden nodes
MAX_TRAIN = 1500 # Maximum number of training sessions
LEARN_RATE = 0.3 # Learning rate
MOMENTUM = 0.1 # momentum factor

random.seed(0)

# Generates random numbers within interval [a, b)
def rand(a, b):
    return (b - a) * random.random() + a

# Generate the matrix of size I * J, the default zero matrix (of course, also use NumPy to speed up)
def makeMatrix(I, J, fill=0.0):
    m = []
    for i in range(I):
        m.append([fill] * J)
    return m

# Function sigmoid, where tanh is used
def sigmoid(x):
    return math.tanh(x)

# Function sigmoid derived function, in order to get the output (ie: y)
def dsigmoid(y):
    return 1.0 - y ** 2

class NN:
    ''' Three - layer back propagation neural network '''

    def __init__(self, ni, nh, no):
        # the node number of Input layer, hidden layer, output layer of
        self.ni = ni + 1 # Add a deviation node
```

```
self.nh = nh
self.no = no

# Activate all nodes of the neural network
self.ai = [1.0] * self.ni
self.ah = [1.0] * self.nh
self.ao = [1.0] * self.no

# Build weight (matrix)
self.wi = makeMatrix(self.ni, self.nh)
self.wo = makeMatrix(self.nh, self.no)

# Set to random value
for i in range(self.ni):
    for j in range(self.nh):
        self.wi[i][j] = rand(-0.2, 0.2)
for j in range(self.nh):
    for k in range(self.no):
        self.wo[j][k] = rand(-2.0, 2.0)

# Finally, the momentum factor (matrix)
self.ci = makeMatrix(self.ni, self.nh)
self.co = makeMatrix(self.nh, self.no)

def update(self, inputs):
    if len(inputs) != self.ni - 1:
        raise ValueError(' Does not match the number of input layer nodes! ')

    # Activate the input layer
    for i in range(self.ni - 1):
        # self.ai[i] = sigmoid(inputs[i])
        self.ai[i] = inputs[i]

    # Activate the hidden layer
    for j in range(self.nh):
        sum = 0.0
        for i in range(self.ni):
            sum = sum + self.ai[i] * self.wi[i][j]
        self.ah[j] = sigmoid(sum)

    # Activate the output layer
```

```

for k in range(self.no):
    sum = 0.0
    for j in range(self.nh):
        sum = sum + self.ah[j] * self.wo[j][k]
    self.ao[k] = sigmoid(sum)

return self.ao[:]

def backPropagate(self, targets, N, M):
    ''' Reverse transmission '''
    if len(targets) != self.no:
        raise ValueError(' Does not match the number of output layer nodes! ')

    # Calculate the output layer error
    output_deltas = [0.0] * self.no
    for k in range(self.no):
        error = targets[k] - self.ao[k]
        output_deltas[k] = dsigmoid(self.ao[k]) * error

    # Calculate the error of the hidden layer
    hidden_deltas = [0.0] * self.nh
    for j in range(self.nh):
        error = 0.0
        for k in range(self.no):
            error = error + output_deltas[k] * self.wo[j][k]
        hidden_deltas[j] = dsigmoid(self.ah[j]) * error

    # Update the output layer weight
    for j in range(self.nh):
        for k in range(self.no):
            change = output_deltas[k] * self.ah[j]
            self.wo[j][k] = self.wo[j][k] + N * change + M * self.co[j][k]
            self.co[j][k] = change
            # print(N*change, M*self.co[j][k])

    # Update the input layer weights
    for i in range(self.ni):
        for j in range(self.nh):
            change = hidden_deltas[j] * self.ai[i]
            self.wi[i][j] = self.wi[i][j] + N * change + M * self.ci[i][j]
            self.ci[i][j] = change

```

```

        # Calculation error
error = 0.0
for k in range(len(targets)):
    error = error + 0.5 * (targets[k] - self.ao[k]) ** 2
return error

def test(self, patterns):
    rst = []
    for p in patterns:
        # Standardized test results
        t0 = self.update(p[0])
        # Anti-standardization
        t1 = t0[0] * (max(p[2]) - min(p[2])) + min(p[2])
        rst.append(t1)
    return rst

def weights(self):
    # print(' Input layer weight:')
    weight_ipt = []
    for i in range(self.ni):
        # print(self.wi[i])
        weight_ipt.append(self.wi[i])
    # print()
    # print(' Output layer weight:')
    weight_opt = []
    for j in range(self.nh):
        # print(self.wo[j])
        weight_opt.append(self.wo[j])
    return [weight_ipt, weight_opt]

def train(self, patterns, iterations=MAX_TRAIN, N=LEARN_RATE, M=MOMENTUM):
    # N: learning rate
    # M: momentum factor
    for i in range(iterations):
        error = 0.0
        for p in patterns:
            inputs = p[0]
            targets = p[1]
            self.update(inputs)
            error = error + self.backPropagate(targets, N, M)
        if i % 100 == 0:

```

```

    print('error %-.5f' % error)

# csv file
def writeCsv(csvName, title_row, list_data):
    with open(csvName, "w", newline="") as datacsv:
        csv_writer = csv.writer(datacsv, dialect="excel")
        csv_writer.writerow(title_row)
        for row in list_data:
            csv_writer.writerow(row)

# Read the training files and organize the data into the training set format [[[input
vector], [output vector]], [[input vector], [output vector]], ..., [[input vector], [ Output
vector]]]
def readFile(filepath):
    # csv_reader = csv.reader(open(filepath, encoding='utf-8'))
    csv_reader = csv.reader(open(filepath))
    detail = []
    for row in csv_reader:
        detail.append(row)

# Split the entire data set into training sets and test sets
train_data_list = []
test_data_list = []
index = 0
for row in range(7, len(detail)):
    ipt = []
    min_0 = 0.0
    max_0 = 0.0
    for index in range(row - 7, row):
        ipt.append(float(detail[index][0]))
        if float(detail[index][0]) < min_0:
            min_0 = float(detail[index][0])
        if float(detail[index][0]) > max_0:
            max_0 = float(detail[index][0])
    opt = [float(detail[row][0])]
    if opt[0] < min_0:
        min_0 = opt[0]
    if opt[0] > max_0:
        max_0 = opt[0]

```

```

    ipt_s = [] # Standardize the input vector
    opt_s = [] # Standardize the output vector
    if max_0 == min_0 and max_0 != 0:
        for t in ipt:
            ipt_s.append(t / max_0)
            opt_s.append(opt[0] / max_0)
    elif max_0 == min_0 and max_0 == 0:
        for t in ipt:

ipt_s.append(t)
    opt_s.append(opt[0])
    else:
        for t in ipt:
            ipt_s.append((t - min_0) / (max_0 - min_0))
            opt_s.append((opt[0] - min_0) / (max_0 - min_0))
    index = index + 1
    if len(detail) - index > 14:
        train_data_list.append([ipt_s, opt_s, ipt, opt])
    else:
        test_data_list.append([ipt_s, opt_s, ipt, opt])

# print(len(result), len(result[0]), len(result[0][0]), len(result[0][1]))

return [train_data_list, test_data_list]

def main():
    fileNames = ['s1', 's2', 's3', 's4', 's5', 's6', 's7', 's8', 's9', 's10']
    predicts = []
    for fileName in fileNames:
        datalist = readFile(fileName + '.csv')
        # training set
        pat = datalist[0]
        # Create a neural network: the input layer has ipt nodes, the hidden layer has hdn
nodes, the output layer has opt nodes
        ipt = len(pat[0][0])
        opt = len(pat[0][1])
        n = NN(ipt, HID_NODE_SIZE, opt)

        n.train(pat)

```

```
predicts.append(n.test(datalist[1]))  
  
# Look at the weight of training (of course, you can consider the weight of the training  
to a long time  
  
# n.weights()  
writeCsv("data\\node-10.csv", fileNames, predicts)  
  
if __name__ == '__main__':  
    main()
```


Customer Flows Forecasts For Shops On Koubei.com

24/10/2016	366	534	105	275	476	108	70	129	946	83
25/10/2016	-81	219	245	-98	-164	52	86	4	10	83
26/10/2016	131	203	336	48	100	393	-301	356	622	83
27/10/2016	307	488	182	194	-935	139	-292	113	38	30447
28/10/2016	-150	-431	634	375	-2109	-11320	-404	372	1296	87488
29/10/2016	-824	573	634	615	-1774	-11320	-404	484	919	87488
30/10/2016	-824	417	634	901	-1548	-11320	800	451	1054	87488
31/10/2016	-598	84	245	620	-1548	-11320	-144	245	1054	87488
SMOreg										
18/10/2016	267	424	164	223	519	377	395	90	449	59
19/10/2016	329	437	181	166	453	326	389	93	371	60
20/10/2016	290	443	127	193	427	345	445	92	472	37
21/10/2016	425	523	189	222	492	448	457	106	516	63
22/10/2016	652	546	187	312	409	483	527	188	526	77
23/10/2016	562	477	147	284	310	452	458	159	453	70
24/10/2016	268	367	137	189	370	286	375	77	416	50
25/10/2016	280	385	174	175	453	297	345	86	350	47
26/10/2016	341	412	163	182	439	298	469	105	370	47
27/10/2016	359	445	150	201	486	344	429	93	390	52
28/10/2016	489	508	202	253	520	385	407	124	401	66
29/10/2016	570	504	185	293	419	443	449	160	362	64
30/10/2016	487	508	151	315	370	386	471	140	396	62
31/10/2016	265	458	162	233	471	298	378	95	312	46
BPNN										
18/10/2016	440	531	230	177	516	383	510	172	538	82
19/10/2016	454	541	236	223	523	379	482	175	530	82
20/10/2016	455	525	220	249	521	382	496	178	577	79
21/10/2016	498	554	233	273	516	387	512	183	582	76
22/10/2016	572	581	233	265	476	396	545	188	607	87
23/10/2016	550	567	195	274	484	396	545	190	581	88
24/10/2016	416	536	171	236	480	380	534	147	563	71
25/10/2016	429	506	167	256	482	368	506	145	541	69
26/10/2016	450	506	170	258	483	369	523	145	537	71
27/10/2016	467	476	170	264	478	377	524	136	573	69
28/10/2016	481	516	170	271	485	391	531	144	595	74
29/10/2016	545	556	170	274	507	395	557	142	604	74
30/10/2016	554	560	210	277	1034	433	561	151	600	70
31/10/2016	451	485	210	265	1043	408	569	165	561	70

Appendix C: weighted average (not including Multilayer Perceptron)

date/shop_id	897	217	315	380	1350	1557	1745	1776	88	145
18/10/2016	7756	11459	353	104	2325	10371	13538	93	5193	131
19/10/2016	13094	9729	763	421	475	3023	75	205	2106	90
20/10/2016	3672	5310	77	177	1205	2143	1521	51	17491	394
21/10/2016	22	5	213	2099	181	1621	551	391	324	1129
22/10/2016	680	17122	1584	3656	15584	648	4250	1691	5765	852
23/10/2016	1967	26766	625	6022	10777	886	13506	7	13886	569
24/10/2016	839	1	1394	2016	6588	25	6021	1077	1349	1238
25/10/2016	220	623	524	2280	6440	44	104	220	203	1536
26/10/2016	1568	1124	311	2784	796	1030	2	830	3969	1178
27/10/2016	7774	1642	600	404	420	160	1561	392	496	1005
28/10/2016	5041	91	199	669	2075	1288	530	24	668	221
29/10/2016	1716	3246	2503	328	446731	1986	5488	4	15988	65
30/10/2016	6914	103	511	981	346089	489	17020	1916	10143	164
31/10/2016	187	1946	35	1989	107178	3821	0	446	694	406