

ORIGINAL RESEARCH ARTICLE

Design and Implementation of Financial Big Data Visualization Analysis Platform

Xi Chen¹, Bo Fan², Jie Zheng¹, Hongyan Cui^{1*}

¹ School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

² A Bank in China Personal Deposit and Investment Department, Beijing 100000, China

ABSTRACT

At present, it has become a hot research field to improve production efficiency and improve life experience through big data analysis. In the process of big data analysis, how to vividly display the results of the analysis is crucial. So, this paper introduces a set of big data visualization analysis platform based on financial field. The platform adopts the MVC system architecture, which is mainly composed of two parts: the background and the front end. The background part is built on the Django framework, and the front end is built with html5, css3, and JavaScript. The chart is rendered by Echarts. The platform can realize the classification of customers' savings potential through bank data, and make portraits of customers with different savings levels. The data analysis results can be dynamically displayed and interact with users in real time, which has practical application significance for improving the bank's precision marketing.

KEYWORDS: Financial big data, Visualization, Classification, Machine learning

Citation: Xi Chen, Bo Fan, Jie Zheng, *et al.* Design and Implementation of Financial Big Data Visualization Analysis Platform. 2019; 3(1): 40-47. DOI: 10.18063/bdci.v3i1.1168

*Correspondence to: Hongyan Cui

Introduction

With the continuous progress of society, we have entered the era of big data. Big data has the following four characteristics: volume, variety, velocity, value^[1]. So on the basis of this, valuable information is mined through various algorithms and visualized interaction is carried out, so as to maximize the value of data. The value becomes especially important. Taking financial big data as an example, it mainly includes static data and dynamic data, of which static data refers to the natural attributes of the population and business data, mainly including the user's name, gender, home address, work unit, *etc.* copy of information and business information handled by the user^[2] such as financial management of the user, credit card holding, *etc.* Due to finance, all businesses need real-name authentication, so the accuracy and usability of this part of data are stronger. Dynamic data refers to user rows. For data, that is, a large amount of data generated by users in the process of using services, such as user's deposit and withdrawal records, consumption records and information such as asset flow direction can be used to analyze users' consumption preferences, build user transfer networks, and mine users.

In order to provide some tailor-made information for users, we need to know more about the resident location and infer the income of users. Services to Increase Financial Institutions' Profits on the Basis of Improving Users

Satisfaction^[2]. In order to maximize the value of data, this paper designs and implements a visual analysis platform for big data. The platform is based on MVC system architecture and mainly comprises a background part and a front part, wherein the background part is built through django framework, and can access data from a database or a text file, analyze the data by using a machine learning method, presenting the analysis results to front-end display and other functions; The front-end part is composed of html5, css3, JavaScript and combinations, which can realize the portrait of user groups and user individuals in multiple dimensions, and is meaningful for banks to find specific user groups and customize special services.

1. Key technologies

In the process of system development, the design pattern of MVC adopts a layered idea, dividing the system design into model, controller and view three modules, each module only need to complete their own responsible content, so you can from the maximum range decoupling in degree. M (model) refers to the model, which is mainly responsible for data processing v (view) refers to a view, mainly is responsible for rendering and presenting the results; C (controller) refers to the controller, which is mainly responsible for the logical connection between the model and the view^[4-5]. The development of this visualization platform adopts the design ideas of MVC. The front-end interface corresponds to the view layer, the relevant part of the database corresponds to the model layer, and the back-end part corresponds to the control layer.

HTML (Hyper Text Markup Language) is also a norm and standard for standardizing web page performance. HTML5 is the fifth version of HTML ,adding many new features to the original version: for example, media playback elements, better support for offline storage, new form controls, *etc.*^[6]. In the development process of the visualization platform, the layout of all elements in the front-end interface is completed through HTML, such as charts, text, buttons, *etc.*

CSS (Cascading Style Sheets) is a meter used to add styles to structured documents Computer Language^[7]. CSS3 is the latest CSS standard, which highlights the idea of modularity in the previous standards and makes development more convenient.^[8] In the process of developing the visualization platform, we first wrote a global CSS file and then referenced it in different pages, thus achieving the purpose of adding decorative effects to HTML.

JavaScript is a programming language that explains execution. JavaScript has the following characteristics: line-by-line interpretation and execution, object-oriented, real-time interaction, cross-platform *etc.*^[9] These characteristics all determine that the language is a developer friendly programming language. In the process of developing the visualization platform, we embed JavaScript code into HTML, so as to link multiple objects in the Web interface and complete the interaction with the Web.

2. System framework

Django is a development framework based on Python, Web and open source. it is an efficient and open source development framework. There are many repeated steps and troublesome parts in the development process of Web and these are not very important to developers, so some professionals have extracted these contents and integrated a development framework. As the visualization platform is a whole system, it needs to display data in the front-end interface and process data in the back-end. In addition, python has great advantages in data processing, so we chose Django framework.

Echarts is a rich chart library based on JavaScript and supports running on various PC devices or mobile devices. It is compatible with current popular browsers, such as Chrome, Firefox, Safari and so on. The charts displayed at the front end of the visualization platform are all presented through Echarts .The development process is as follows: First, select appropriate charts according to requirements, such as scatter chart, column chart, radar chart, *etc.*; Then set the chart according to Echarts API, such as title, legend, coordinate axis, *etc.* Finally, analyze the data transmitted in the background and show it.

JQuery is a simple, efficient and very popular JavaScript framework. JQuery's consistent design concept is "write less, do more", which advocates writing less code and realizing more functions^[11]. From this point, we can also see that

jQuery pays great attention to efficiency. The core features of jQuery are as follows: object-oriented, plug-in extension mechanism is perfect, code portability is strong, *etc.*^[12] For the development of this visualization platform, jQuery is indispensable, and many operations need jQuery to complete. Take click response as an example. First, we need to add click events inside HTML and then define click functions through jQuery to respond, such as jumping, turning pages, hiding, etc.

3. Demand analysis

The purpose of this system is to build a user-friendly data visualization analysis platform, and its design style conforms to the current application for a wide range of Web systems. In order to get better display effect, the system adopts B-S (Browser-Server) Design Architecture, Not C-S(Client-Server) architecture. The system adopts a modular design idea and divides the system into three modules: data storage module and algorithm An analysis module and a front-end display module, wherein the data storage module is responsible for realizing the data access function, and the algorithm analysis module is negative responsible for using machine learning methods to mine valuable information in the data, and the front-end display module is responsible for displaying the analysis results. The sample greatly reduces the coupling inside the system and provides convenience for the development and maintenance of the system. On the whole, the system can be divided into two parts: front-end and back-end. The following will analyze the system requirements from these two parts.

3.1 Background demand analysis

3.1.1 Data access

For data visualization analysis platform, data is indispensable, so data access is a problem that must be faced. In this part of data access, data from different sources should be accessed. The commonly used data sources include the following three types: for scenes with small data volume, a single file is usually directly used for storage for convenience; For scenes with large amount of data, such as TB and level of data, if the data processing demand is not high, it is usually stored in a database, and the more widely used ones are Mysql and Oracle; If the demand for data processing is high, Hive is usually used to store data. Hive is a distributed relational database that can be used to process large amounts of data in parallel and distributed manner. As the data processing part of this project focuses on algorithm analysis, the data access module does not need too much data processing, so when the amount of table data is small, use csv files for access, and when the amount of table data is large, use oracle for access.

3.1.2 Algorithm analysis

For the whole data visualization analysis platform, the proposed analysis algorithm is an important part that affects the display effect of the whole platform. In the joint adjustment of visualization system and algorithm, firstly, this part will realize the pretreatment of the data read in the previous step: for unsupervised machine learning methods, such as clustering, unified dimensional processing operations such as normalization and standardization are required. As for supervised machine learning methods, such as classification and regression, need to process the original data to obtain training sets and test sets, *etc.*^[13] Next, this part will propose the corresponding algorithm model according to the specific scientific problems to be solved, or call the existing set well done Scikit-Learn algorithm package. Finally, the output analysis results of the algorithm model are displayed on the visualization platform.

3.1.3 Front and rear end communication

In the development process of the whole system, we have always adopted the idea of modularity, developing front-end and back-end separately. The coupling between systems is minimized, but front-end and back-end communication is the key part to be solved in the whole development process. The first is after Taiwan to Front End: All display effects of the front end interface are based on the data of the back end, so the data is in the appropriate grid. It is especially important to transmit the data to the front end, which determines the stability of the system. The second is front-end to back-end, because the system is designed with modules the idea of blocking, so the front end is only used

for data display, all processing operations are carried out in the background, then the user requests made at the front end should be transmitted to the background in an appropriate format so as to process and respond to user requests.

3.2 Front end requirement analysis

3.2.1 User group level

At the user group level, firstly, the system should be able to divide users into different potential users according to the analysis results of clustering algorithm and user groups are displayed. This is the first interface for users to enter the system, and it is the result exhibition of primary classification of user groups. Secondly, on the basis of the first-level classification, users should be subdivided and the results of the second-level classification should be displayed. Finally, the needle. For each secondary classification of the user group, the deposit time distribution of each category and the average and maximum value of the user's deposit value should be entered line display, in order to more targeted service users.

3.2.2 Individual user level

At the level of individual users, firstly, for a certain user in the user group, the 12-month savings value of the user should be added line display; Secondly, the regression prediction algorithm is used to predict the savings value of the user in the last month, and the prediction result and prediction are mistaken. Difference is displayed in the system. Finally, based on the user's consumption record data, the consumption situation is divided into pos machine, WeChat and Alipay for statistics. The total consumption value and consumption times of each consumption mode are counted and displayed in the personal interface of the platform.

4. Specific implementation

4.1 Background implementation

The backstage part is mainly implemented based on the framework of django. On the premise of ensuring the availability and stability of the system, it is proposed high development efficiency. The schematic diagram of the background module is shown in **Figure 1**.

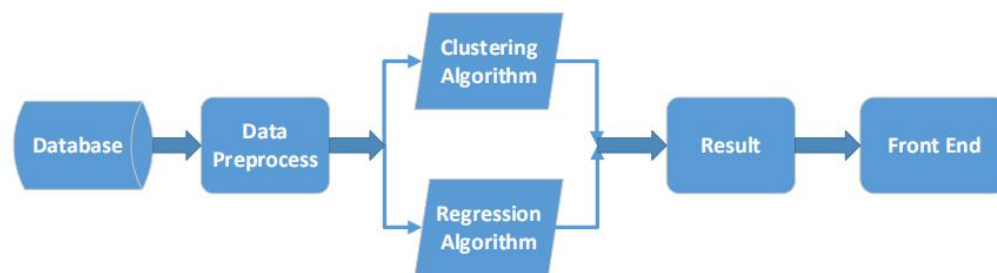


Figure 1. Schematic diagram of background module

4.1.1 Data access module

In order to improve the compatibility of the system, we have designed two ways to access the data. The first method is based on text piece by piece, take csv file as examples. After configuring the file path information, use the open and readline commands to import the csv file and close the file. One thing to note in this process is that the data read from the csv file is not the data format we want, but the data in string format, and the data in each row are linked together. The row data needs to be split using the split command and then converted into the int type. The second way is to access data for the database. The first is the database connection. The database information needs to be configured in the settings.py file, including the database name, user name, password, host ip address and port number. The database information of this system is configured based on the oracle and database of the laboratory. The details are as follows:

JDBC: Oracle: THIN: @ *. *. *: 1521: database name. Next, you need to write sql code to access the data. In order to facilitate the development, django has integrated sql statements for common operations such as adding, deleting, modifying and querying the database. Only modifying the field information and query conditions in the sql statements can complete the data access operation.

4.1.2 Algorithm analysis module

In order to further improve the function of the system, the algorithm analysis part is implemented in the background. The algorithm analysis part is mainly through to The implementation of the next two sub-parts: firstly, it is based on Elkan k-means algorithm, which is used to divide the user group, thus aiming at not portrait with the same user group. This algorithm is an unsupervised clustering algorithm, using clustering error SSE to measure the clustering effect. Results, so as to select the best clustering category; The second is based on SVR algorithm, which is used to predict the user's savings value. The calculation method is a supervised machine learning method, so the data set needs to be divided into two parts: training and testing by using the set aside method. The concrete implementation of the set-aside method is accomplished through train_test_split in sklearn. Finally, the prediction effect of the algorithm is measured at the individual level and the overall level respectively.

4.1.3 Front and rear communication modules

This part realizes the function of front-end and back-end communication. First, configure the settings.py file in the django framework. It mainly includes setting project path, setting front-end code path, setting network parameters, configuring database, etc. This document is django is a very important file in the framework, and most of the setting operations of the framework are based on this file. Second, configure url.py file. This is a file that defines the url and mapping rules .Only by adding url and mapping information to this file can the background respond to the url and request of the front end. In addition, import the daemon entry through url.py file, that is, Source.view file. Finally, a data transmission class is implemented to convert the data analyzed by the algorithm into json format for transmission. This is a widely used data transmission format at present, especially in front-end and back-end communication.

4.2 Implementation of front end

The front end part is mainly constructed by combining html5, css3, JavaScript and jQuery is adopted in the construction process.The display of frames and charts is based on Echarts and Chart Library. Each page in the front end will be shown below.

4.2.1 User group display

Firstly, according to the user's savings value, the user group is divided into four categories based on our proposed clustering algorithm, which are stable respectively fixed user group, active user group, potential user group and lost user group. Compared with other users. The three categories account for a large proportion, so in order to make the display effect more intuitive, we will be active user group, potential user group and loss. In addition, pie charts have been added to these three categories of users to display details, which is convenient for managers to query details interactively. The displayed system interfaces are as **Figure 2** shows following.

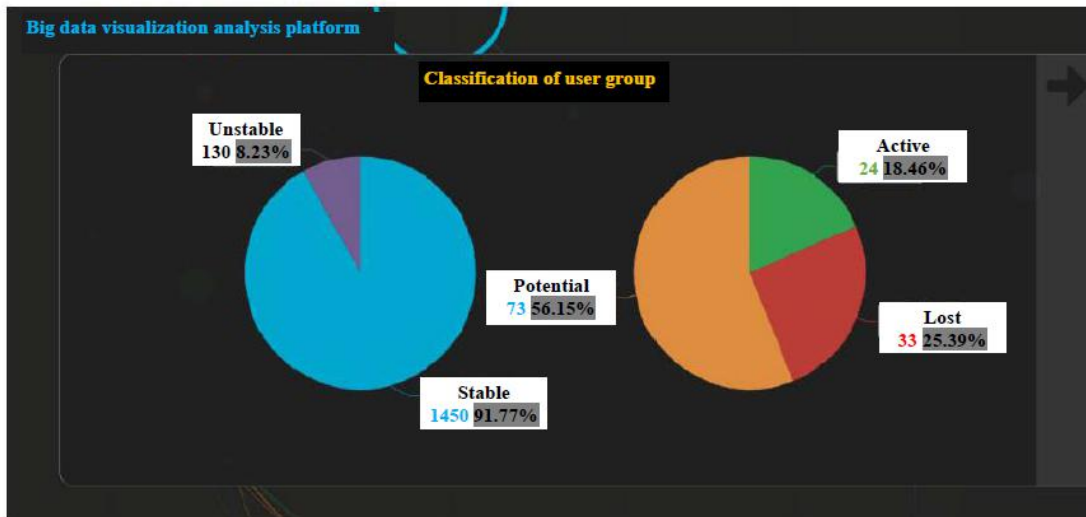


Figure 2. Level 1 classification interface of user group

In the user group level 1 classification interface, the pie chart on the left shows the population of stable user groups and other three user groups. The pie chart on the right shows the proportion of active users, potential users and lost users.

Next, in order to further analyze the user groups, clustering is continued on the basis of the four types of user groups. The algorithm is divided, in which stable user groups are divided into 5 and 2-level classified user groups, which are stable 1 to stable 5 respectively; Active user groups are divided into 2 and 2 secondary classified user groups, corresponding to active 1, active 2; The churn user group is divided into 2 and 2 sub-class user groups, corresponding to churn 1, churn 2. Potential user groups are divided into 2 and 2 secondary classification user groups, corresponding to potential 1, potential 2. Then we will draw a portrait of the user group with secondary classification, as shown in Figure 3.



Figure 3. User group secondary classification portrait interface

In the Figure 3 of User Group Secondary Classification Portrait Interface (taking stable user group as an example in the figure), the left side shows the proportion of user group secondary classification, of which stable type, 1 has the highest proportion and stable type 4 has the lowest proportion. On the right is the average savings level of the secondary classified user group. Here we show the average savings value through three indicators, namely, the average savings value of April, 2017 to March, 2018. It can be seen from the figure that for a specific stable user group, the values of these three indexes are very close, which also reflects the characteristics of stable user group from the side. On the other hand, the average level of different stable user groups is also different, from stable type 1 to stable type 5 the average

level is gradually increasing, which also shows that the secondary classification of stable user groups is very meaningful. The lower side shows the distribution of the average savings and the maximum savings of all secondary users, with black indicating the maximum savings and white indicating the average savings.

4.2.2 Individual user display

Finally is the personal user portrait interface, where the representative users in the user group are selected for display (Users shown in the figure belong to fluctuating user groups). On the one hand, the user's portrait information predicts the user's savings value through a prediction algorithm and the calculation prediction error is obtained, on the other hand, based on statistical information. The display interface is shown in **Figure 4**.

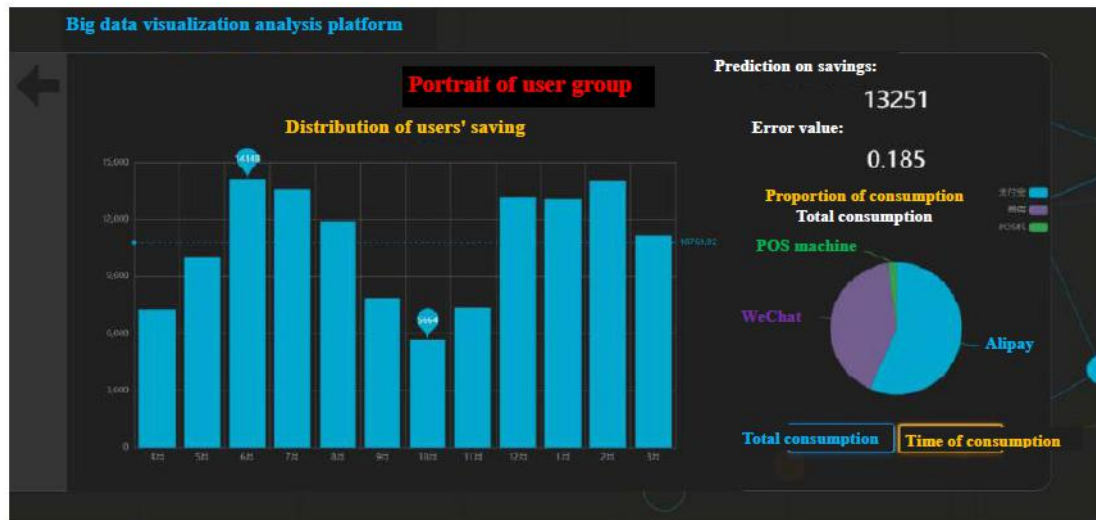


Figure 4. Personal user portrait interface

In the personal user portrait interface, the left side shows the distribution map of the user's savings value. We have selected 12 months of ample user's savings value from April, 2017 to March, 2018 to be displayed. It can be seen from the figure that the fluctuation of the user's savings value is still relatively large, reaching the maximum value of 14148 yuan in June 2017 and the minimum value of 5664 yuan in October 2017, which is in line with the characteristics of fluctuating user groups. The upper right corner shows the savings value predicted by the prediction algorithm and the relative prediction error, that is, the ratio of the absolute value of the error value to the real value. The lower right corner shows the consumption proportion of users. First, three widely used consumption modes are selected: Alipay, WeChat and POS. Then, the display is carried out through two dimensions: consumption times and total consumption value. As can be seen from fig. 4, the user's Alipay consumption accounts for a relatively high proportion, followed by wechat, and POS machine accounts for the lowest proportion, which is also very consistent with the consumption situation in real life.

5. Conclusion

This paper introduces the design and implementation of the big data visualization analysis platform. First of all, the design and development process of the system. The key technologies and system framework used in are introduced, mainly including MVC and design framework and use django, jQuery, Echarts and Framework. Next, the requirements of the system are analyzed, which is completed through the background function requirement analysis and front-end function requirement analysis. Finally, the system is implemented on the basis of function requirement analysis. The back-end part of the system can realize the functions of data access, algorithm analysis, front-end and back-end communication, etc. The front-end can be used for specific purposes. The two subjects, the user group and the individual user can be drawn pictures from multiple angles, thus realizing a visual analysis platform that can interact with the user and be friendly to the user.

Acknowledgment

This paper is guided by experts in relevant fields and supported by the Ministry of Education-China Mobile Research Fund Project (No.: MCM20170306) and the project of Research on the Boundary and Strategy of Personal Information Protection in Big Data Era.

References

1. Fan J, Han F, Liu H. Challenges of big data analysis. *National Science Review* 2014; 1(2): 293-314.
2. Natal IDP, Garcia ACB. Activity recognition model based on gps data, points of interest and user profile//International symposium on methodologies for intelligent systems. Springer, Cham 2017; 358-367.
3. Zhang K. Implementation scheme of mobile phone user portrait on big data platform. *Information Communication* 2014; (2).
4. Ladd S, Davison D, Devijver S, et al. *Expert spring MVC and web flow*. Berkeley, CA: Apress; 2006.
5. Liu J, Dai J. Research of lightweight web application based on spring MVC and iBATIS frameworks. *Journal of Computer Applications* 2006; 4: 26.
6. Lubbers P, Albers B, Salim F, et al. *Pro HTML5 programming*. New York, NY, USA: Apress; 2011.
7. Frain B. *Responsive web design with HTML5 and CSS3*. Packt Publishing Ltd; 2012.
8. McFarland DS. *CSS3: The missing manual*. "O'Reilly Media, Inc."; 2012.
9. Bray T. *The javascript object notation (json) data interchange format*. 2017.
10. Forcier J, Bissex P, Chun WJ. *Python web development with Django*. Addison-Wesley Professional, 2008.
11. Chaffer J, Swedberg K. *Learning jQuery[M]*. Packt Publishing Ltd, 2011.
12. Nixon R. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. "O'Reilly Media, Inc.", 2014.
13. Sagioglu S, Sinanc D. Big data: A review//2013 International Conference on Collaboration Technologies and Systems (CTS). IEEE, 2013: 42-47.