2020 2nd International Conference on Advanced Control, Automation and Artificial Intelligence (ACAAI 2020)
ISBN: 978-1-60595-673-2

Research on Application of Data Visualization in Finance

Yu ZHANG, Hou SHU*, Hua-qun LIU, Shi-jie WANG and Xin-zhe ZHANG

Beijing Institute of Graphic Communication, Beijing, China

*Corresponding author

Abstract. Finance and data visualization are inseparable, and its processes are data acquisition, data analysis, and data generation. Data acquisition is data acquisition through crawlers and other methods or open interfaces of financial data; the process of data analysis is mainly to select relevant financial models for data processing; data generation is data visualization. The visualization of financial data externalizes the analysis process of financial data, and displays complex and large data groups in the form of charts to further discover the value of the data.

Process and Technology of Financial Data Visualization

Based on the development of big data in recent years, data visualization in related fields has become particularly important. The design of data flow makes the process of finding rules from data easier and more regular. Based on this, the financial field the data visualization application has also been further developed. By analyzing the financial data left by previous transactions, it can help people accurately find crisis and potential opportunities from the data, and also make the decision-making perspective in the financial field more scientific[1].

Big data visualization has formed a quantifiable process, and its process can be roughly divided into three stages: acquisition, processing, and generation. Based on the particularity of financial data and the characteristics of big data, the process of financial data visualization can be summarized into three stages: data acquisition, data analysis, and data generation[2].

Data Collection

Data acquisition is the first stage of financial data analysis. At this stage, the industry background of financial data needs to be clarified, available reference cases should be compiled based on existing data, and appropriate financial data models should be selected for analysis based on the goals set. Data analysis indicators and related charts. Currently available data is divided into structured data and unstructured data. The main contents of the two are shown in Table 1.

Table 1. Data classification table.

	Structured data	Unstructured data
historical data	Daily closing price	Financial Article
Real-time data	Bid/ask prices for stocks	Real-time financial news
Raw data	Commission/Transaction, Account Fund	Public opinion, announcement, research report
Processed data	Minutes/hours/days/weeks/ months/years	Relevance

In order to ensure the integrity and privacy of financial data, the financial data required in the analysis process is obtained here through an open network interface. The most commonly used Python language in big data analysis is used to introduce the package used to obtain financial data in Python. Use Yahoo's financial data open interface to obtain data in the following format:

```
import pandas as pd

import numpy as np

import pandas_datareader as pdr
```

Get the stock data of a company (using Sohu as an example), the format is as follows:

```
sohu = pdr.get_data_yahoo ('SOHU', start = start)
```

Data Analysis

The process at this stage mainly includes three processes: data cleaning, data storage, and selection of appropriate models. When the first phase of data analysis is completed, a structured data set will be obtained. The first step requires data cleaning, that is, the process of removing data such as errors and redundancy from the obtained data set according to the needs of the target. The second step is data storage. The cleaned canonical data set is stored in a locally created database; the third step is to select a suitable model for data analysis based on the target you want to analyze.

Data Cleaning

First determine whether there is a missing value in this data, and then determine the range of missing values. You can make separate strategies based on the missing proportion and field importance. When there is less missing data, you can selectively re-obtain the data to avoid years with missing data; when there is more missing data, you can try to obtain data from other open interfaces or use crawlers to obtain unstructured data.

Missing Value Processing. First determine whether there is a missing value in this data, and then determine the range of missing values. You can make separate strategies based on the missing proportion and field importance. When there is less missing data, you can selectively re-obtain the data to avoid years with missing data; when there is more missing data, you can try to obtain data from other open interfaces or use crawlers to obtain unstructured data.

Outlier Processing. Outliers generally include three types of values: incorrect values, format errors, and logical errors. Value errors include range errors and bit errors. There are mainly three types of format errors. The display formats of time, date, value, and half-width are inconsistent; there are characters that should not exist in the content; the content does not match the content of the field. Logical error cleaning mainly includes removing or replacing unreasonable data values.

Data Storage

Data storage is to save the formatted data set to a data format that can be used by data visualization tools to facilitate subsequent analysis and visualization. According to the format required by the selected visualization tool, it can be stored in CSV, json, or Excel format in general. The processed data set is stored as a new database to avoid the unstable network interface and possible occurrence of data during use. The problem. Save the obtained data to the local in CSV/xlsx/json format (optional one), the format is as follows:

```
data = pd.read_csv ('sohu.csv', index_col = 0, parse_dates = True)

data.head ()

data.to_csv ('sohu.csv')

data.to_excel ('sohu.xlsx')

data.to_json ('sohu.json')
```

Select Model

With the development of big data, many important financial models have been derived in the financial field, such as portfolio theory, capital asset pricing models, efficient market assumptions, and option pricing theories. These models are based on the normal distribution of stock data. Select the appropriate financial model and process it according to the results of analysis.

Data Visualization

The generation of data is the process of data visualization. At this stage, the processed basic data set is used to establish a data model that meets the goals[3], and the data is visualized through visual coding. There are many ways to display data visualization, which can be roughly divided into static display and dynamic (interactive) display. Static display can draw the analyzed results into charts, word clouds, models, etc., and appear in the form of pictures or web pages. The characteristic is to collect data at a certain time interval, and then clean up the latest collected data and then import it into the total data set, so as to achieve the effect of real-time update. Due to the dynamic display of the hugeness of the data set, the display focus is usually displayed in different charts. Classification, users can observe the results of real-time changes through interactive operations.

Application of Financial Data Visualization-stock Trend Prediction

Take Alibaba's open stock data as an example for visual process demonstration.

Demand Analysis

For time series such as finance, big data has its own unique analysis methods. The most commonly used is the ARIMA model[4]. It is a model that makes predictions based on time series. It is often used for demand forecasting and in planning, the idea of the model is to learn the pattern that changes with time from historical data, and use it to predict the future when you learn it. Based on the open stock data since Alibaba's listing, it analyzes stock trends, and establishes a differential integrated moving average autoregressive model (ARIMA model[5]) to predict the differences between Alibaba's stock real trends from January 2019 to March 2019.

In this case, the ARIMA model is selected. The basic steps of modeling are as follows.

- Obtaining time series data of the observed system.
- Draw the data into a graph and observe whether the graph is stable; if the graph is not stable, first perform a d-order difference operation to process the data into a stationary time series;
- The auto-correlation coefficient ACF and partial auto-correlation coefficient PACF are respectively obtained from the stabilized data, and the graphics are obtained. By analyzing the auto-correlation graph and the partial auto-correlation graph, the best level p and order q are obtained;

- After the results d, q, and p of the above three steps are processed, an ARIMA model is obtained, and then the model is tested.

Data Visualization Process

The analysis and visualization of stock data conforms to the general data visualization process. In order to ensure the integrity of the data, the stock data port opened by Yahoo (https://sg.finance.yahoo.com) is used to obtain Alibaba's listed stock data. Since Yahoo's open data is already clean data, the data cleaning operation is omitted here. The time of data collection is set to early 2009 to January 4, 2019. Matplotlib using python is used to analyze Alibaba's stock data. The overall process is shown in Figure 1.

Code

- Import the packages required for stock analysis, set the style required for picture display, specify Chinese fonts, determine the time period of the stock data to be processed, and set the time range here from January 1, 2009 to April 1, 2019 , Read Alibaba stock data through the interface and preview, the code is as follows.

```
import pandas as pd

import pandas_datareader

import datetime

import matplotlib.pylab as plt

from matplotlib.pylab import style

from statsmodels.tsa.arima_model import ARIMA

from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Set the theme style of the picture display

style.use ('ggplot')

# Solve the problem that matplotlib displays Chinese

plt.rcParams ['font.sans-serif'] = ['SimHei'] # specify the default font

plt.rcParams ['axes.unicode_minus'] = False # Solve the problem that the saved image is a negative sign '-' and displayed as a square

# 1. Prepare the data

# Specify stock analysis start date

start_date = datetime.datetime (2009, 1, 1)

# Specify stock analysis deadline

end_date = datetime.datetime (2019, 4, 1)

# Stock Symbol

stock_code = 'BABA' # Alibaba

stock_df = pandas_datareader.data.DataReader (
```

stock_code, 'yahoo', start_date, end_date

)

Preview data

print (stock_df.head ())

The preview is shown in Figure 2.

Figure 1. Stock processing flowchart.

Date	High	Low	Open	Close	Volume	Adj Cl
2014-09-19	99.699997	89.949997	92.699997	93.889999	271879400	93.889
2014-09-22	92.949997	89.500000	92.699997	89.889999	66657800	89.889
2014-09-23	90.480003	86.620003	88.940002	87.169998	39009800	87.169
2014-09-24	90.570000	87.220001	88.470001	90.570000	32088000	90.570
2014-09-25	91.500000	88.500000	91.089996	88.919998	28598000	88.919

Figure 2. Alibaba stock preview.

- Visualize the collected stock data, the code is as follows.

plt.plot (stock_df ['Close'])

plt.title ('Daily Closing Price')

Plt.show()

The visualization is shown in Figure 3.

- The stock data is resampled by week and displayed visually, the code is as follows.

stock_s = stock_df ['Close']. resample ('W-MON'). mean ()

stock_train = stock_s ['2014': '2018']

plt.plot (stock_train)

plt.title ('Weekly Closing Stock Price')

Plt.show()

The visualization is shown in Figure 4.

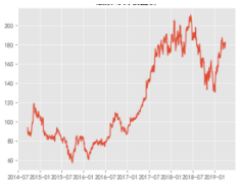

Figure 3. Daily closing price of the stock.

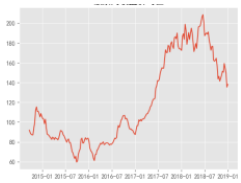

Figure 4. Weekly closing price.

- Analyze the autocorrelation coefficient ACF and partial autocorrelation coefficient PACF of the stock, the code is as follows.

acf = plot_acf (stock_train, lags = 20)

plt.title ("ACF for Stock Index")

acf.show ()

pacf = plot_pacf (stock_train, lags = 20)

plt.title ("PACF for Stock Index")

pacf.show ()

The visualization is shown in Figure 5 and Figure 6.

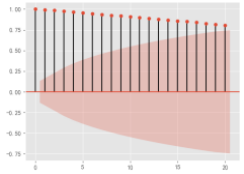

Figure 5. Stock Index ACF.

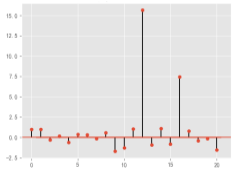

Figure 6. Stock Index PACF.

- Use a section to process stock data to smooth the time series. The code is as follows.

stock_diff = stock_train.diff ()

diff = stock_diff.dropna ()

print (diff.head ())

print (diff.dtypes)

plt.figure ()

plt.plot (diff)

plt.title ('First Order Difference')

Plt.show()

acf_diff = plot_acf (diff, lags = 20)

plt.title ("ACF of first order difference")

acf_diff.show ()

pacf_diff = plot_pacf (diff, lags = 20)

plt.title ("PACF of the first order difference")

pacf_diff.show ()

Plt.show()

The visualization is shown in Figure 7, Figure 8 and Figure 9.

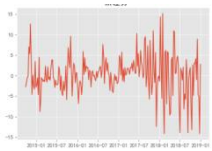

Figure 7. First order difference.

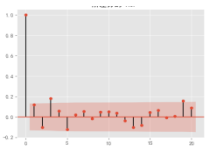

Figure 8. First-order differential ACF.

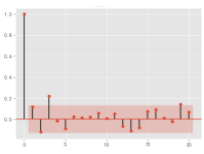

Figure 9. First-order differential PACF.

- Based on the ACF and PACF ranking and fitting the ARIMA model, the stock data trend from January 2019 to March 2019 is predicted, and the visualization is shown in Figure 7.

```
model = ARIMA (stock_train, order = (1, 1, 1), freq = 'W-MON')
arima_result = model.fit ()
print (arima_result.summary ())
pred_vals = arima_result.predict (start = str ('2019-01'), end = str ('2019-03'),
                                  dynamic = False, typ = 'levels')
print (pred_vals)
stock_forcast = pd.concat ([stock_s, pred_vals], axis = 1, keys = ['original', 'predicted'])
plt.figure ()
plt.plot (stock_forcast)
plt.title ('Real vs. Predicted')
Plt.show()
```

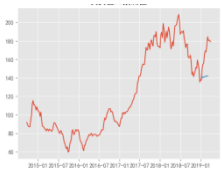

Figure 10. Real value/predicted value.

Conclusion

It can be seen from the visualization that the predicted value of Alibaba stock trend has a certain deviation from the real value. There may be two reasons for this analysis. The first is that the processed stock data undergoes only first-order difference processing, and the processing process is relatively simple. The obtained data has a certain deviation, so the real value and the predicted value have a large deviation. The second is that the selected model is relatively single, and several more models should be selected for comparison and processing to obtain comprehensive results.

Acknowledgments

Based on the development of data science, the process design of data visualization makes the process of discovering rules from data more convenient and regular, and makes people's perspective more scientific. However, due to the high complexity of data, different processing methods will lead to different results of data visualization. It can be seen in the data visualization of the stock correlation analysis that the data visualization can more clearly show the rules between financial data, and use time series models to predict the results. Due to the different data processing methods selected, and in this application only using a method, the data results may be biased, so the accuracy of the results cannot be guaranteed, and subsequent verification studies are needed.

References

[1] D. Garlaschelli and M. I. Loffredo. *Structure and evolution of the world trade network*. Physica A: Statistical Mechanics and its Applications, 355(1):138–144, Sept. 2005.

[2] Liang Jiye, Feng Chenjiao, Song Peng. *Review of Big Data Correlation Analysis*. Chinese Journal of Computers, 2016.1

[3] Ren Lei, Du Yi, Ma Shuai, Zhang Xiaolong, Dai Guozhong. *Overview of Big Data Visual Analysis*. Journal of Software, 2014, 25 (9): 1909-1936.

[4] Yang Junjie, Liao Zhuofan, Feng Chaochao. *Research on VaR Model of Financial Time Series and Construction of Web Visualization System* .East China Jiaotong University. 2015.

[5] Liu Yiming. *Research on Financial Time Series Forecasting Model*. Lanzhou University, 2014.