Stock Prediction System

Renjie Zhang

November 16, 2017

1 Revision History

Date	Version	Notes
2017-10-06	1.0	create
2017-10-09	1.1	update

2 Symbols, Abbreviations and Acronyms

symbol	description
T	Test
SVM	Support Vector Machine
RDD	Resilient Distributed Datasets
K	Kernel function
X	features of the data (date , price)
C	The price from different days
β	The error modifier
y	The result between 1 and -1

Contents

List of Tables

List of Figures

3 General Information

This Document is a test plan of the software-Stock Prediction System which is a tool used to predict the stock prices using machine learning.

[The text is better for version control, and for reading in other editors, if you use a hard-wrap at 80 characters —SS]

3.1 Purpose

This test plan describes the testing methods that will drive the testing of the Stock Prediction System and give a guide to the users about the QA. This document includes the descriptions of testing tool, testing functions, unit testing method and system test. Based on these testing, the users may cache the functional and non functional issues from the first release of the software and find out the improvement.

3.2 Scope

The testing plan will cover both system testing and unit testing. The functions are the data input, data format validation, calculation of SVM, data plot, Spark RDD Distributed System, and the output of the result. Basically every part of the system will be tested.In [proof read —SS] this plan, most of the testing will be done by unit testing.

[An explicit web-link to your GitHub repo would be nice. —SS]

[Explicitly state programming language —SS]

[Reference your SRS document —SS]

3.3 Overview of Document

This document explains the testing plan for the software - Stock Prediction System. It includes the briefcase of software description, the different testing methods - unit testing and integration testing, testing tools. It will cover both functional and non functional requirements.

4 Plan

In this section, the general description of the testing plan for the software and corresponding testing tools will be introduced.

4.1 Software Description

The Stock Prediction System is used to analyze the future trend of stocks. The prediction was provided by machine learning algorithms based on the historical data. The system will be run on a big data platform (Spark), in order to [replace "in order to" with "to" —SS] obtain the more accurate results. In this case, we need to setup a distributed system to support Spark.

4.2 Test Team

Renjie Zhang

4.3 Automated Testing Approach

NA

4.4 Verification Tools

Python unit testing : This framework was included in Python standard library. It is easy to use by people familiar with the xUnit frameworks, strong support for test organization and reuse via test suites
Wing : A popular Python IDE which contents the code checking and indent correction.
PyChecker (Optional) : A source checking tools which finds problems that are typically caught by a compiler for less dynamic languages; imports each module before checking it.
[good idea to use PyChecker —SS]

4.5 Non-Testing Based Verification

NA

5 System Test Description

This section describes the testing on the system environment, such as the data file input and spark distribution system. Users need to make sure the application run on the system without any issues.

5.1 Tests for Functional Requirements

The functional requirements includes the data input, the data format verification, the Spark RDD transaction, the plot generation and the display of the output. The software must fulfill all of the functional requirements without issues.

5.1.1 Data Input

The system needs to load a dataset from a CSV file. This test ensures the data input method.

File loading Testing

1. File is loaded successfully

   Type: Functional, Manual, Static etc. Initial State: NA Input: File Path Output: Successful Message How test will be performed: System tries to load the data set file based on the file name and location without issues.

2. Input Data Validation

   Type: Functional, Manual Initial State: NA Input: Data from the file Output: Successful message How test will be performed: System have to ensure the data type and format is correct for each column of the file such as the pattern of the date, the formats of the price and number of digits of decimals. Shown in Figure 1 [End sentences with a period. —SS] [Use cross-referencing in LaTeX, rather than hard coding the values. —SS] [You shouldn't be saying "such as" in your test description. You should describe everything that you are looking for. What do you do if the data is in the incorrect format? What could go wrong? Where are the test cases to ensure that your software behaves correctly in those situations? —SS]

   [Your test description is not specific enough. What will the contents of the file b? If the data is difficult to specify, maybe you could point to a sample file in your repo that you are going to use? Your test plan document could then point to the file in the repo. —SS]

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	8/24/2017	957.42	959	941.14	952.45	952.45	5195700
3	8/25/2017	956	957.62	944.1	945.26	945.26	3324800
4	8/28/2017	946.54	953	942.25	946.02	946.02	2596700
5	8/29/2017	940	956	936.33	954.06	954.06	2874300
6	8/30/2017	958.44	969.41	956.91	967.59	967.59	2904600
7	8/31/2017	974.7	981	972.76	980.6	980.6	3331500
8	9/1/2017	984.2	984.5	976.88	978.25	978.25	2535900
9	9/5/2017	975.4	976.77	960.37	965.27	965.27	2883200
10	9/6/2017	968.32	971.84	960.6	967.8	967.8	2129900
11	9/7/2017	974	980.59	972.55	979.47	979.47	2566800
12	9/8/2017	979.1	979.88	963.47	965.9	965.9	2583300
13	9/11/2017	974.46	981.94	974.22	977.96	977.96	2186700

Figure 1:

5.1.2 Distributed System

This section focus on the distributed system of Spark platform. It guides the users to test the data flow on the spark system. RDD is the format of data flow in Spark. testers need to double check the data was distributed into each workers through RDD.

[This description doesn't have enough detail. What is the array that will be sent to each worker? What is the mark? Provide enough information that someone (other than you) could write the test. —SS]

Spark RDD Testing

1. Data is distributed to workers

   Type: Functional, Manual, Static etc. [You shouldn't use etc. here —SS] Initial State: NA Input: An Array from the driver Output: An updated array with marks of each worker How test will be performed: There will be an input array to the driver, driver assign the array to each workers. Workers add a mark with its identification to elements of the array and return it back to driver. Shown in Figure 2.

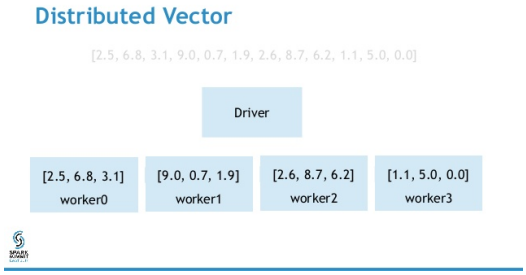

Figure 2:

5.1.3 Testing on The Algorithm

This section gives the guide to the testing on the Support Vector machine algorithms. It covers the plot test and the calculation of functions in SVM.

SVM Testing

1. Plot Testing

   Type: Functional, Manual, Static etc. Initial State: NA Input: A pre-defined Array Output: A correct plot How test will be performed: System reads the data and generate a plot using the price and the date as the x and y axis.

2. SVM Calculation

   Type: Functional, Manual, Static etc. Initial State: NA Input: A collection of data Output: correct results How test will be performed: The detail of the functions related to SVM Calculation can be found in Unit Test

[If SVM is a unit test, then it shouldn't also appear here. —SS]

5.2 Tests for Non functional Requirements

Non functional requirements are not as prior as functional requirements, but it is still necessary to have tests on them in order to improve the software performance and quality.

5.2.1 Big Data System Testing

In this section, the only requirement is the performance.

Performance Testing

1. Performance of the workers

   Type: Manual Initial State: NA Input/Condition: NA Output/Result: Result of the time cost
   How test will be performed: It is a good way to increase the performance by increase the number of works. Try to increase one more worker and compare the time cost on data training and testing.

[Not enough detail. How many workers are you going to experiment with? What test case are you going to get them to perform? What are the axes on the graph you are going to produce as output of the testing? —SS]

5.3 Traceability Between Test Cases and Requirements

	Input Data	Plot	Data Validation	Calculation	Output	Performanc
Input Data testing	X					
Data Validation	X		X			
RDD Testing	X					
Plot testing		X				X
Kernelling				X		
Volatility Testing				X		
Momentum Testing				X		
Prediction Testing				X	X	
Performance Testing						X

Table 1: Traceability Matrix Showing the Connections Between Requirements and Test Cases

[If you use shorter names for the column headings, it will fit better on the page. —SS]

6 Unit Testing Plan

The unit testing plan will involve the following modules: Load Data, SVM Kernelling, Volatility Calculating, Momentum Calculating, Predict and Output.

1. Load Data
   The detail of the data loading was explained on 5.1.1

2. Kernelling
   In machine learning, kernel methods are a class of algorithms for pattern analysis, whose best known member is the support vector machine (SVM) RBF Kernel is used for price forecasting. A correct result is expected by inputting the parameters to the equation
   $K\left(X_i, X_k\right) = \exp\left(-\frac{1}{\delta^2}\sum_{n}^{j=1}(X_{ij} - X_{kj})^2\right)$

3. Calculate Volatility
   This function is use to calculate the price volatility and index volatility. Since the calculation is similar they share the same function. A correct result is expected by inputting the parameters to the equation
   $\frac{\sum_{i=t-n+1}^{t} \frac{C_i - C_{i-1}}{Ci-1}}{n}$

4. Calculate Momentum
   This function is use to calculate the price momentum and index momentum. Since the calculation is similar they share the same function. A correct result is expected by inputting the parameters to the equation
   $\frac{\sum_{i=t-n+1}^{t} \frac{C_i - C_{i-1}}{Ci-1}}{n}$

5. Predict
   The Predict module receives a set of parameters calculated from the previous functions and returns a result. A correct result is expected by inputing the parameters to the equation

   $y = \beta_0 + \sum a_i y_i K(x(i), x)$

7 Appendix

NA

7.1 Symbolic Parameters

NA

7.2 Usability Survey Questions?

NA