

# Stock Prediction System

Renjie Zhang

October 19, 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

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>General Information</b>	<b>1</b>
3.1	Purpose . . . . .	1
3.2	Scope . . . . .	1
3.3	Overview of Document . . . . .	1
<b>4</b>	<b>Plan</b>	<b>2</b>
4.1	Software Description . . . . .	2
4.2	Test Team . . . . .	2
4.3	Automated Testing Approach . . . . .	2
4.4	Verification Tools . . . . .	2
4.5	Non-Testing Based Verification . . . . .	2
<b>5</b>	<b>System Test Description</b>	<b>3</b>
5.1	Tests for Functional Requirements . . . . .	3
5.1.1	Data Input . . . . .	3
5.1.2	Distributed System . . . . .	4
5.1.3	Testing on The Algorithm . . . . .	4
5.2	Tests for Nonfunctional Requirements . . . . .	5
5.2.1	Big Data System Testing . . . . .	5
5.3	Traceability Between Test Cases and Requirements . . . . .	5
<b>6</b>	<b>Unit Testing Plan</b>	<b>5</b>
<b>7</b>	<b>Appendix</b>	<b>7</b>
7.1	Symbolic Parameters . . . . .	7
7.2	Usability Survey Questions? . . . . .	7

## List of Tables

## List of Figures

This document ...

## **3 General Information**

### **3.1 Purpose**

This test plan describes the testing methods that will drive the testing of the Stock Prediction System. The document introduces:

Test Strategy: rules the test will be based on, including the givens of the project (e.g.: start / end dates, objectives, assumptions); description of the process to set up a valid test (e.g.: entry / exit criteria, creation of test cases, specific tasks to perform, scheduling, data strategy). Execution Strategy: describes how the test will be performed and process to identify and report defects, and to fix and implement fixes. Test Management: process to handle the logistics of the test and all the events that come up during execution (e.g.: communications, escalation procedures, risk and mitigation, team roster)

### **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.

### **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

### 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 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 contains the code checking and indent correction.

PyChecker : A source checking tools which finds problems that are typically caught by a compiler for less dynamic languages; imports each module before checking it.

[Thoughts on what tools to use, such as the following: unit testing framework, valgrind, static analyzer, make, continuous integration, test coverage tool, etc. —SS]

### 4.5 Non-Testing Based Verification

Code inspection : The code inspection includes spelling error and syntax error. Since the programming language is Python, the indentation checking is required. Those errors could be detected by the tools Wing and PyChecker.

Code walkthrough : The code walkthrough is used to find out the run time errors such as the array index range, divided by 0 and exception handling.

[List any approaches like code inspection, code walkthrough, symbolic execution etc. Enter not applicable if that is the case. —SS]

## 5 System Test Description

### 5.1 Tests for Functional Requirements

#### 5.1.1 Data Input

##### Testing for file loading

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 formate of the price and number of digits of decimals.

### **5.1.2 Distributed System**

#### **Spark RDD Testing**

1. Data is distributed to workers

Type: Functional, Manual, Static etc.

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.

### **5.1.3 Testing on The Algorithm**

#### **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



## 5.2 Tests for Nonfunctional Requirements

### 5.2.1 Big Data System Testing

#### 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.

## 5.3 Traceability Between Test Cases and Requirements

## 6 Unit Testing Plan

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

1. Load Data

The detail of the data loding 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 inputing the parameters to the equation

$$k(X_i, X_k) = \exp\left(-\frac{1}{\delta^2} \sum_{n=1}^j (X_{ij} - X_{kj})^2\right)$$

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 inputing the parameters to the equation

$$\frac{\sum_{i=t-n+1}^t \frac{C_i - C_{i-1}}{C_{i-1}}}{n}$$

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 inputing the parameters to the equation

$$\frac{\sum_{i=t-n+1}^t \frac{C_i - C_{i-1}}{C_{i-1}}}{n}$$

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

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

[Unit testing plans for internal functions and, if appropriate, output files —SS]

## **7 Appendix**

This is where you can place additional information.

### **7.1 Symbolic Parameters**

The definition of the test cases will call for SYMBOLIC\_CONSTANTS. Their values are defined in this section for easy maintenance.

### **7.2 Usability Survey Questions?**

This is a section that would be appropriate for some teams.