

Project Title

Author Name

October 2, 2017

Contents

1	Reference Material	ii
1.1	Table of Units	ii
1.2	Table of Symbols	ii
1.3	Abbreviations and Acronyms	iii
2	Introduction	1
2.1	Purpose of Document	1
2.2	Scope of Requirements	1
2.3	Characteristics of Intended Reader	1
2.4	Organization of Document	1
3	General System Description	1
3.1	System Context	1
3.2	User Characteristics	2
3.3	System Constraints	2
4	Specific System Description	2
4.1	Problem Description	2
4.1.1	Terminology and Definitions	2
4.1.2	Physical System Description	3
4.1.3	Goal Statements	3
4.2	Solution Characteristics Specification	3
4.2.1	Assumptions	4
4.2.2	Theoretical Models	4
4.2.3	General Definitions	4
4.2.4	Data Definitions	5
4.2.5	Instance Models	6
4.2.6	Data Constraints	7
4.2.7	Properties of a Correct Solution	8

5	Requirements	8
5.1	Functional Requirements	8
5.2	Nonfunctional Requirements	9
6	Likely Changes	9
7	Traceability Matrices and Graphs	9
8	Appendix	13
8.1	Symbolic Parameters	13

1 Reference Material

This section records information for easy reference.

1.1 Table of Units

Throughout this document SI (Système International d’Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
m	length	metre
kg	mass	kilogram
s	time	second
°C	temperature	centigrade
J	energy	Joule
W	power	Watt ($W = J s^{-1}$)

[Only include the units that your SRS actually uses —SS]

1.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	unit	description
A_C	m^2	coil surface area
A_{in}	m^2	surface area over which heat is transferred in

[Use your problems actual symbols. The si package is a good idea to use for units. —SS]

1.3 Abbreviations and Acronyms

symbol	description
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
Stock Prediction System	[put your program name here —SS]
T	Theoretical Model

[Add any other abbreviations or acronyms that you add —SS]

2 Introduction

The Stock Prediction system is used to analysis the future trend of stocks. The prediction was provided by machine learning algorithms based on the historical data. The system will run on a big data platform (Spark), in order to obtain the more accury results. In this case, we need to setup a distributed system to support Spark.

2.1 Purpose of Document

The purpose of this document is to explain how to implement a whole system for machine learning prediction. Stock is the example being used here. With different algorithms and libraries, the system can be used for more types of predictions. In this case, we will use support vector machine althrithsm to predict the future trend of stocks based on the historical data.

2.2 Scope of Requirements

The purpose of this software is to give the user a reference by calculating the possibility of the future price change. For example, it may go up with a chance of xx percent and go down with a chance of xx percent.

2.3 Characteristics of Intended Reader

Reader are not required to have any specific knowledge background. However, it is very helpful to have some basic knowledge of big data and machine learning.

2.4 Organization of Document

This document will cover the configuration of the big data platform, the work flow of the program and the algorithm of the stock and the machine learning.

3 General System Description

The project have an interface to let user choose the company but on the demo there will be only one company to display. A char graph will be draw by the system as well. The diagram displays the historical data and future trend posibility.

3.1 System Context

- User Responsibilities:
 - Choose the stock they wish to predict
 - Decide the date time range of the historical period

- Update the historical data set
- Stock Prediction System Responsibilities:
 - Load data set from files and display errors when the loading fails
 - Display the plot of the stock
 - Predict the future trend based on the historical data

3.2 User Characteristics

The user of the system should have some basic knowledge of machine learning algorithms (Support vector Machine).

3.3 System Constraints

The system is run on Linux type operating system, since it is the ideal OS for Spark.

4 Specific System Description

This system is used to predict the future trend of stocks based on the historical data from Yahoo Finance using Support Vector Machine algorithm.

4.1 Problem Description

Stock Prediction System is a tool to predict stocks based on machine learning and big data. Stock prediction is not a new technology, but the result is not that satisfied. One of the most important reason is the historical data is not supportive due to the hardware and software limitation. With the development of bigdata technology, machine learning prediction have obtained a significant success.

4.1.1 Terminology and Definitions

- Support Vector Machine A supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis.
- Spark A big data platform that save and retrieve data from different machines in a format called RDD(Resilient Distributed Datasets). It provides a set of machine learning libraries and support different types of programming languages.
- RDD Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.

- **Distributed System** A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages. It combines a set of computers to work on a single task as one machine.

4.1.2 Physical System Description

The physical system of Stock Prediction System, as shown in Figure ?, includes the following elements:

- **Driver** The drive is the machine that send requests to the works and receive the response from. Data will be converted into RDD format and equally transfer to each work. Drive itself does not do the actual works.
- **Work** The works are the machines which do the actual jobs. They receive data from drive using RDD and return an RDD back to drive after certain processes.

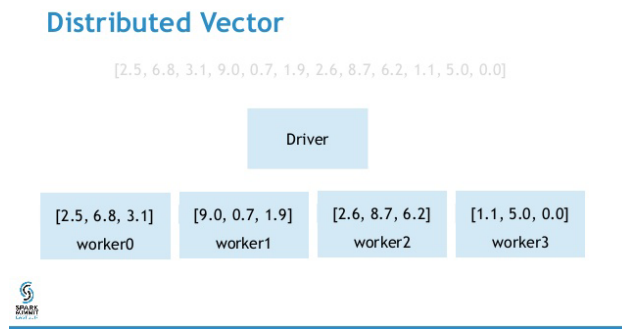


Figure 1

4.1.3 Goal Statements

- **Predicture Result1:]** A result of the future stock price with its probability.

4.2 Solution Characteristics Specification

The instance models that govern Stock Prediction System are presented in Subsection 4.2.5. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

4.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

- A1: [Short description of each assumption. Each assumption should have a meaningful label. Use cross-references to identify the appropriate traceability to T, GD, DD etc., using commands like dref, ddref etc. —SS]

4.2.2 Theoretical Models

This section focuses on the general equations and laws that Stock Prediction System is based on. [Modify the examples below for your problem, and add additional models as appropriate. —SS]

Number	T1
Label	Support Vector Machine
Equation	$-\nabla \cdot \mathbf{q} + g = \rho C \frac{\partial T}{\partial t}$
Description	The above equation gives the conservation of energy for transient heat transfer in a material of specific heat capacity C ($\text{J kg}^{-1} \text{ } ^\circ\text{C}^{-1}$) and density ρ (kg m^{-3}), where \mathbf{q} is the thermal flux vector (W m^{-2}), g is the volumetric heat generation (W m^{-3}), T is the temperature ($^\circ\text{C}$), t is time (s), and ∇ is the gradient operator. For this equation to apply, other forms of energy, such as mechanical energy, are assumed to be negligible in the system (A??). In general, the material properties (ρ and C) depend on temperature.
Source	http://www.efunda.com/formulae/heat_transfer/conduction/overview_cond.cfm
Ref. By	GD??

4.2.3 General Definitions

This section collects the laws and equations that will be used in deriving the data definitions, which in turn are used to build the instance models. [Some projects may not have any content for this section, but the section heading should be kept. —SS] [Modify the examples below for your problem, and add additional definitions as appropriate. —SS]

Number	GD1
Label	Newton’s law of cooling
SI Units	W m^{-2}
Equation	$q(t) = h\Delta T(t)$
Description	<p>Newton’s law of cooling describes convective cooling from a surface. The law is stated as: the rate of heat loss from a body is proportional to the difference in temperatures between the body and its surroundings.</p> <p>$q(t)$ is the thermal flux (W m^{-2}).</p> <p>h is the heat transfer coefficient, assumed independent of T (A??) ($\text{W m}^{-2} \text{ }^{\circ}\text{C}^{-1}$).</p> <p>$\Delta T(t) = T(t) - T_{\text{env}}(t)$ is the time-dependent thermal gradient between the environment and the object ($^{\circ}\text{C}$).</p>
Source	(?, p. 8)
Ref. By	DD1, DD??

Detailed derivation of simplified rate of change of temperature

[This may be necessary when the necessary information does not fit in the description field. —SS]

4.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. [Modify the examples below for your problem, and add additional definitions as appropriate. —SS]

Number	DD1
Label	Heat flux out of coil
Symbol	q_C
SI Units	W m^{-2}
Equation	$q_C(t) = h_C(T_C - T_W(t))$, over area A_C
Description	T_C is the temperature of the coil ($^{\circ}\text{C}$). T_W is the temperature of the water ($^{\circ}\text{C}$). The heat flux out of the coil, q_C (W m^{-2}), is found by assuming that Newton's Law of Cooling applies (A??). This law (GD1) is used on the surface of the coil, which has area A_C (m^2) and heat transfer coefficient h_C ($\text{W m}^{-2} ^{\circ}\text{C}^{-1}$). This equation assumes that the temperature of the coil is constant over time (A??) and that it does not vary along the length of the coil (A??).
Sources	?
Ref. By	IM1

4.2.5 Instance Models

This section transforms the problem defined in Section 4.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 4.2.4 to replace the abstract symbols in the models identified in Sections 4.2.2 and 4.2.3.

The goals [reference your goals —SS] are solved by [reference your instance models —SS]. [other details, with cross-references where appropriate. —SS] [Modify the examples below for your problem, and add additional models as appropriate. —SS]

Number	IM1
Label	Energy balance on water to find T_W
Input	$m_W, C_W, h_C, A_C, h_P, A_P, t_{\text{final}}, T_C, T_{\text{init}}, T_P(t)$ from IM?? The input is constrained so that $T_{\text{init}} \leq T_C$ (A??)
Output	$T_W(t), 0 \leq t \leq t_{\text{final}}$, such that $\frac{dT_W}{dt} = \frac{1}{\tau_W}[(T_C - T_W(t)) + \eta(T_P(t) - T_W(t))]$, $T_W(0) = T_P(0) = T_{\text{init}}$ (A??) and $T_P(t)$ from IM??
Description	T_W is the water temperature ($^{\circ}\text{C}$). T_P is the PCM temperature ($^{\circ}\text{C}$). T_C is the coil temperature ($^{\circ}\text{C}$). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^{\circ}\text{C}$, where 0°C and 100°C are the melting and boiling points of water, respectively (A??, A??).
Sources	?
Ref. By	IM??

Derivation of ...

[May be necessary to include this subsection in some cases. —SS]

4.2.6 Data Constraints

Tables 2 and 4 show the data constraints on the input and output variables, respectively. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 2 are listed in Table 3.

(*) [you might need to add some notes or clarifications —SS]

Table 2: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
L	$L > 0$	$L_{\min} \leq L \leq L_{\max}$	1.5 m	10%

Table 3: Specification Parameter Values

Var	Value
L_{\min}	0.1 m

Table 4: Output Variables

Var	Physical Constraints
T_W	$T_{\text{init}} \leq T_W \leq T_C$ (by A??)

4.2.7 Properties of a Correct Solution

A correct solution must exhibit [\[fill in the details —SS\]](#)

5 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

5.1 Functional Requirements

- R1: [\[Requirements for the inputs that are supplied by the user. This information has to be explicit. —SS\]](#)
- R2: [\[It isn't always required, but often echoing the inputs as part of the output is a good idea. —SS\]](#)
- R3: [\[Calculation related requirements. —SS\]](#)

R4: [Verification related requirements. —SS]

R5: [Output related requirements. —SS]

5.2 Nonfunctional Requirements

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —SS]

6 Likely Changes

LC1: [Give the likely changes, with a reference to the related assumption (aref), as appropriate. —SS]

7 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an “X” may have to be modified as well. Table 5 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 6 shows the dependencies of instance models, requirements, and data constraints on each other. Table 7 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

[You will have to modify these tables for your problem. —SS]

The purpose of the traceability graphs is also to provide easy references on what has to be additionally modified if a certain component is changed. The arrows in the graphs represent dependencies. The component at the tail of an arrow is depended on by the component at the head of that arrow. Therefore, if a component is changed, the components that it points to should also be changed. Figure ?? shows the dependencies of theoretical models, general definitions, data definitions, instance models, likely changes, and assumptions on each other. Figure ?? shows the dependencies of instance models, requirements, and data constraints on each other.

	T1	T??	T??	GD1	GD??	DD1	DD??	DD??	DD??	IM1	IM??	IM??	IM??
T1													
T??			X										
T??													
GD1													
GD??	X												
DD1				X									
DD??				X									
DD??													
DD??								X					
IM1					X	X	X				X		
IM??					X		X		X	X			X
IM??		X											
IM??		X	X				X	X	X		X		

Table 5: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM??	IM??	IM??	4.2.6	R??	R??
IM1		X				X	X
IM??	X			X		X	X
IM??						X	X
IM??		X				X	X
R??							
R??						X	
R??					X		
R2	X	X				X	X
R??	X						
R??		X					
R??			X				
R??				X			
R4			X	X			
R??		X					
R??		X					

Table 6: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??	A??
T1	X																		
T??																			
T??																			
GD1		X																	
GD??			X	X	X	X													
DD1							X	X	X										
DD??			X	X						X									
DD??																			
DD??																			
IM1											X	X		X	X	X			X
IM??												X	X			X	X	X	
IM??														X					X
IM??													X					X	
LC??				X															
LC??								X											
LC??									X										
LC??											X								
LC??												X							
LC??															X				

Table 7: Traceability Matrix Showing the Connections Between Assumptions and Other Items

References

8 Appendix

[Your report may require an appendix. For instance, this is a good point to show the values of the symbolic parameters introduced in the report. —SS]

8.1 Symbolic Parameters

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —SS]