CS2310 Final Report

Feature Selection using Ontology for SIS Predictive Cancer Detection Component

Jeongmin Lee
jel158@pitt.edu

Link to Youtube video: https://www.youtube.com/watch?v=tY5kzPCBDLA
Link to Sourcefile:
https://www.dropbox.com/sh/igdjc8a2j8n4iim/AACfbWZU2XBkQlCNkB3iXSq8a?dl=0

1. Introduction and motivation

Predictive and personalized medicine is emerging as critical field in health care area. For Slow Intelligent System (SIS) in personal health care system, here I am suggesting the new component that is capable of providing cancer prediction service based on analyzing each patient's gene (microarray expression) data. The main technology used by new component is the classification method in machine learning area and the major challenge of classification involving gene dataset is that there exist few samples but very large number of features so that it is hard to build a robust classifier that can solve overfitting, the problem known as curse of dimensionality in machine learning field.

To resolve this issue, a number of feature selection and extraction methods are suggested and here, in this project, we are suggesting to use Gene Ontology for feature selection and to make use of this approach toward predictive cancer detection component in the SIS personalized health care system.

2. Internal Architecture

System Overview and Setting

The system is composed of Health Care Manager (HCM) that connects all components of the system. The our proposing Predictive Cancer Detection Component (PCDC) is connected to the HCM so that existing components in SIS personalized health care system can interact with PCDC through HCM.

Within PCDC, there are several components working together to conduct its mission. Following is detail description of each component.

PCDC Manager

PCDC Manager is working as a supercomponent within PCDC component. It provides access point for each sub component of PCDC and facilitates communications between each subcomponent. It provides GUI to user so that user can interact and control the PCDC through this manager program.

Cancer Classifier Module (CCM)
Before prediction (classification) service is provided, the classifier should be trained by the training dataset that consisted of cancerous and non-cancerous people's microarray expression gene dataset and their labeling. The training step needs user's parameters such as which type of classifier to use and configuration of the classifier such as thresholds for each classification algorithms. Also, since our training classifier leverages existing knowledge of interactions of each gene by using Gene Ontology (GO), the CCM also connected to GO to do the training task.

Cancer Dataset Database (CDD)
The CCD contains all gene samples and their labels of each patient registered at our SIS personalized health care system and also datasets of publicly available cancer gene samples for classifier training task.

Gene Ontology (GO)
Gene Ontology is publicly provided by the Gene Ontology Consortium (www.geneontology.org) . Although it is operating externally from our system, we consider the GO inside of our component since it is integral part of our PCDC. The GO provides API so that our system can access it through internet.

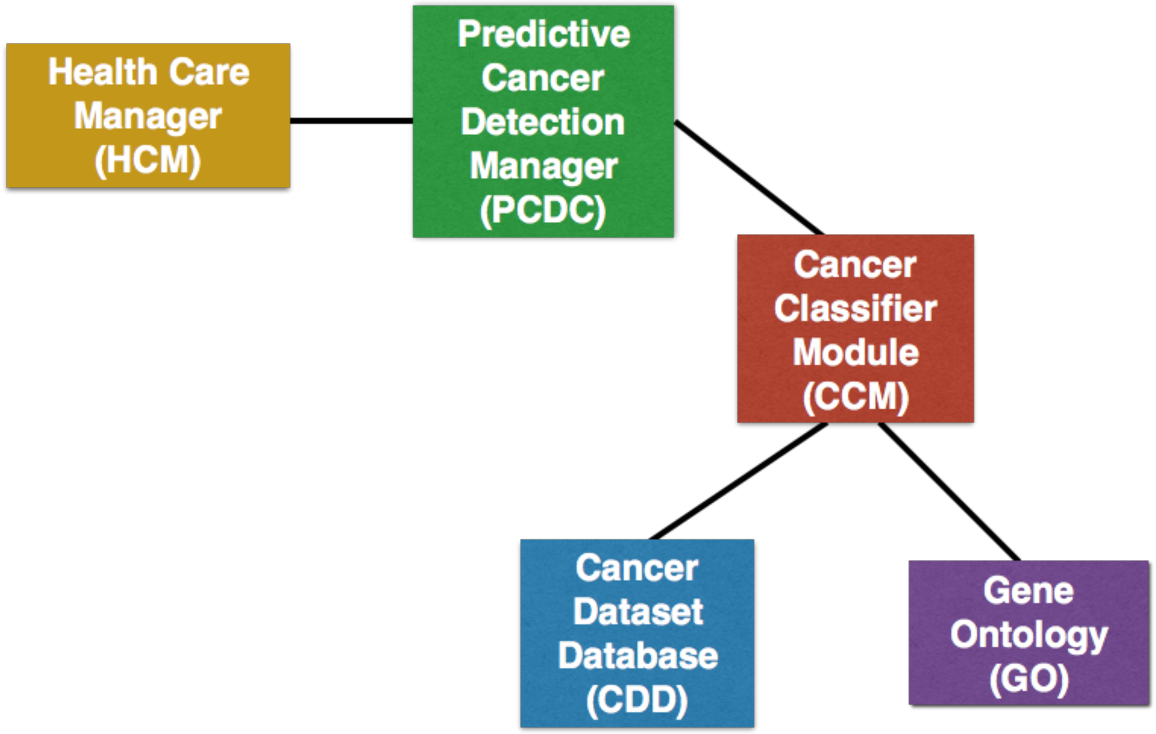

Health Care Manager (HCM)
Predictive Cancer Detection Manager (PCDC)
Cancer Classifier Module (CCM)
Cancer Dataset Database (CDD)
Gene Ontology (GO)

Figure1: Diagram of system overview

3. Component Working Procedure

Overview
The working procedure of PCDC is comprised of two steps. At first step, it trains cancer detection classifier and it happens by CCM involving CDD and GO and the user (presumably a cancer detection system operator/researcher) interacts and controls the system through PCDC Manager.

Once classifier is trained, the user (also a cancer detection system operator/researcher or an autonomous program) can make a prediction of patient's status of cancer based on the patient's microarray expression (gene) data.

Step1: Build (train) Cancer Detection Classifier
1) A user initiates this step by selecting and choosing type of classifier. The PCDC provides GUI and gets user input and send message of requesting the build stage to CCM. Following is choices user can select for the classifier training task.

Type of Classifier
- Logistic Regression
- Support Vector Machine
- Stacked Autoencoder (Deep Neural Network)
- Principal Component Analysis

User Parameter that provided by each component
- Numeric data

2) Once request is come to the Cancer Classifier Module (CCM), the CCM interacting with Cancer Dataset Database (CCD) to retrieve training dataset of publicly available microarray expression dataset and Gene Ontology (GO) to retrieve the gene ontology dataset. Then, it creates classifier by training the dataset with given configuration of user provided.
Once building classifier is done, it also conducts validating created model and report the result back to PCDC Manager by sending message.

3) Depending on the result from 2), the 1) can be retrained by step through 1) and 2) nd the decision is made by the user. Once the user confirms with current classifier, the step1 is completed.

Step2: Detect Cancer by Classifier

1) Once step1 is completed, now user can start to predict patient's status of cancer or the patient can get cancer or not based on the patient's microarray expression (gene) data. A user can start this task by sending gene data of a patient from PCDC Manager to CCM.

2) When CCM gets request of classify, it classify the result based on the patient's gene data and send back the result contained within a message to PCDC Manager.

4. Specification of Messages

Overview

- 1) Messages for registration of each component and subcomponent
- 2) Messages for building classifier
    - Initiate the task; message containing parameters from PCDC Manager to CCM.
    - From CCM to CCD to retrieve sample dataset for training and its response.
    - From CCM to GO to retrieve gene ontology data and its response.

- 3) Messages for detecting cancer by classifier
    - Initiate the task; message from PCDC Manager to CCM with patient's gene data.
    - From CCM to PCDC Manager, a message of result of the classification.

Messages

- 1) Registration of each component and subcomponent
    - xml/InitXML/CreateCancerClassifierModule.xml
    - xml/InitXML/CreateCancerDatasetDatabase.xml
    - xml/InitXML/CreateGeneOntology.xml
    - xml/InitXML/CreatePCDCManager.xml

- 2) Build classifier
    - Initiate the task; message containing parameters from PCDC Manager to CCM.
        - xml/DataXML/CancerClassifierModuleConnect.xml
        - xml/DataXML/CancerClassifierModule

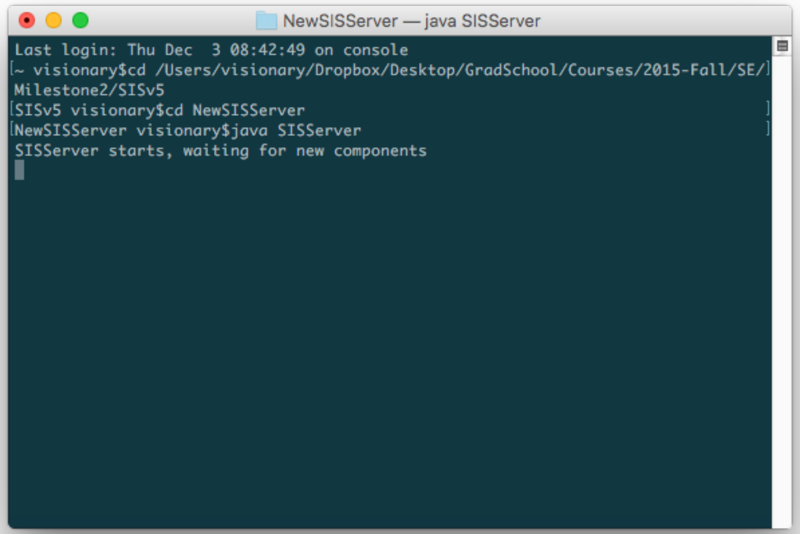
    - From CCM to CCD to retrieve sample dataset for training and its response.
    - From CCM to GO to retrieve gene ontology data and its response.
    - From CCM to PCDCManager to notify training classifier is done

- 3) Detect Cancer by Classifier
    - Initiate the task; message from PCDC Manager to CCM with patient's gene data.
    - From CCM to PCDC Manager, a message of result of the classification.

5. Implementation and operating components

Step1. Run SIS Server

To initiate the whole system, the first key step is to run SIS Server. The SIS Server is located at NEWSISServer folder and when it is run, it will be looks like following figure.

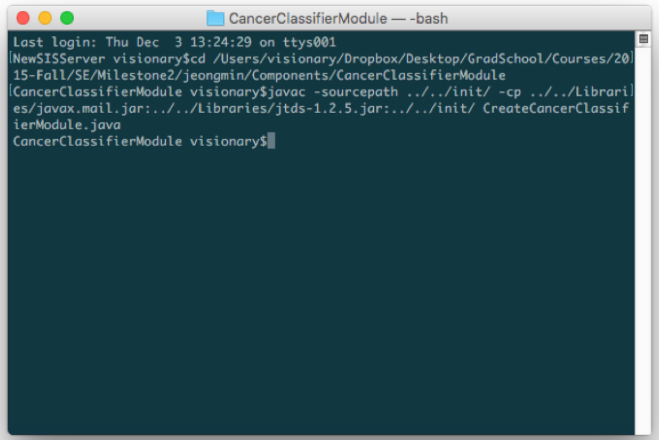


```
Last login: Thu Dec  3 08:42:49 on console
[~ visionary$cd /Users/visionary/Dropbox/Desktop/GradSchool/Courses/2015-Fall/SE/]
Milestone2/SISv5
[SISv5 visionary$cd NewSISServer                                                 ]
[NewSISServer visionary$java SISServer                                           ]
SISServer starts, waiting for new components
```

Figure: Running SISSerer

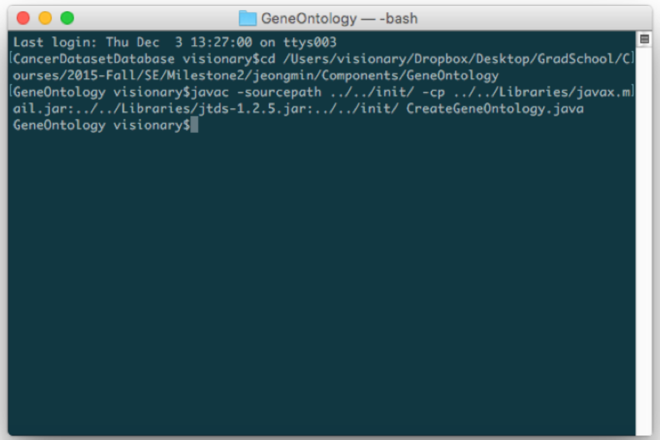
Step2. Compile each Component Except Super Component
Please note that super component can be compiled after other components are compiled.

1) **Cancer Classifier Module**
   at Components/CancerClassifierModule
   javac -sourcepath ../../init/ -cp ../../Libraries/javax.mail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreateCancerClassifierModule.java



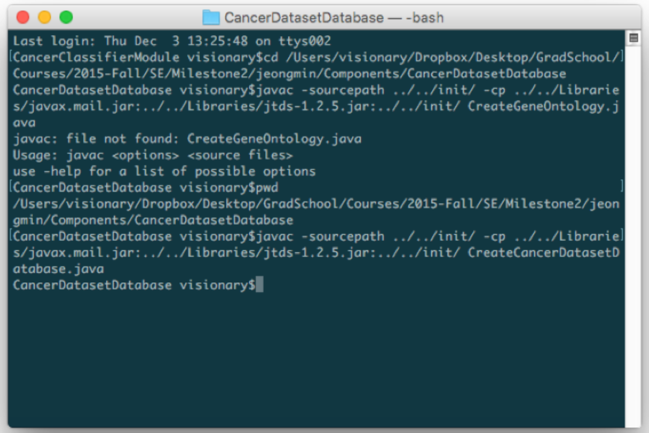
```
Last login: Thu Dec  3 13:24:29 on ttys001
[NewSISServer visionary$cd /Users/visionary/Dropbox/Desktop/GradSchool/Courses/20]
15-Fall/SE/Milestone2/jeongmin/Components/CancerClassifierModule
[CancerClassifierModule visionary$javac -sourcepath ../../init/ -cp ../../Librari]
es/javax.mail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreateCancerClassif
ierModule.java
CancerClassifierModule visionary$
```

2) **Gene Ontology**
   at Components/GeneOntology
   javac -sourcepath ../../init/ -cp ../../Libraries/javax.mail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreateGeneOntology.java



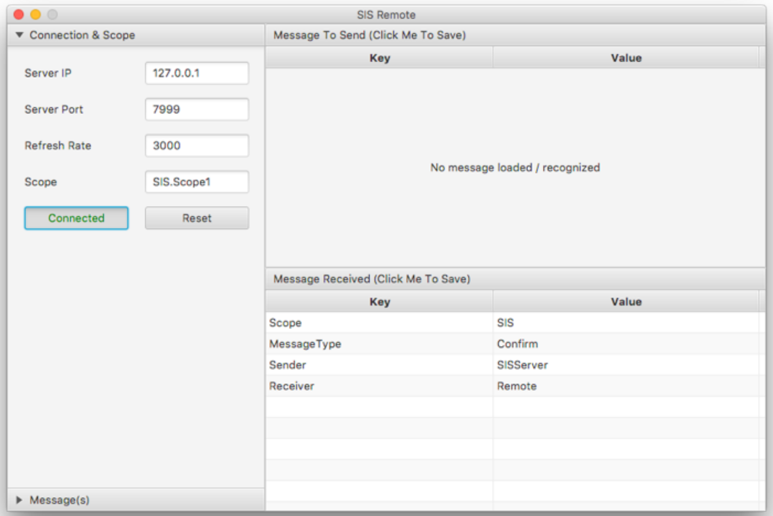
```
Last login: Thu Dec  3 13:27:00 on ttys003
CancerDatasetDatabase visionary$cd /Users/visionary/Dropbox/Desktop/GradSchool/C
ourses/2015-Fall/SE/Milestone2/jeongmin/Components/GeneOntology
GeneOntology visionary$javac -sourcepath ../../init/ -cp ../../Libraries/javax.m
ail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreateGeneOntology.java
GeneOntology visionary$
```

3) **Cancer Dataset Database**
   at Components/CancerDatasetDatabase
   javac -sourcepath ../../init/ -cp ../../Libraries/javax.mail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreateCancerDatasetDatabase.java



```
Last login: Thu Dec  3 13:25:48 on ttys002
CancerClassifierModule visionary$cd /Users/visionary/Dropbox/Desktop/GradSchool/
Courses/2015-Fall/SE/Milestone2/jeongmin/Components/CancerDatasetDatabase
CancerDatasetDatabase visionary$javac -sourcepath ../../init/ -cp ../../Librarie
s/javax.mail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreateGeneOntology.j
ava
javac: file not found: CreateGeneOntology.java
Usage: javac <options> <source files>
use -help for a list of possible options
CancerDatasetDatabase visionary$pwd
/Users/visionary/Dropbox/Desktop/GradSchool/Courses/2015-Fall/SE/Milestone2/jeon
gmin/Components/CancerDatasetDatabase
CancerDatasetDatabase visionary$javac -sourcepath ../../init/ -cp ../../Librarie
s/javax.mail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreateCancerDatasetD
atabase.java
CancerDatasetDatabase visionary$
```

Step3. Run PrjRemote and Connect to SIS Server
In our proposed prototype system, since PrjRemote has the role of debugger and user interface as well, it is important to operate PrjRemote Properly. Once it is initiated and connected to SIS Server, it will be looks like following figure.


SIS Remote
Connection & Scope
Message To Send (Click Me To Save)
Message Received (Click Me To Save)
Message(s)

Setting	Value
Server IP	127.0.0.1
Server Port	7999
Refresh Rate	3000
Scope	SIS.Scope1

Connected | Reset

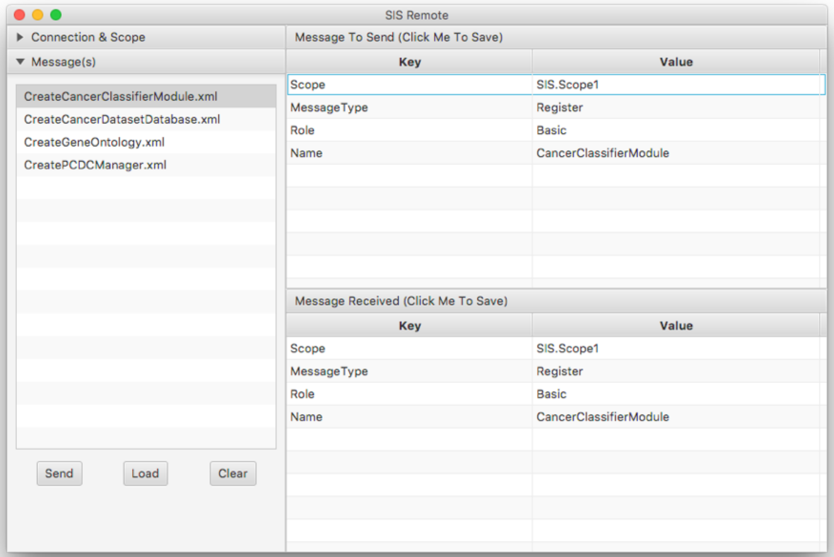
No message loaded / recognized

Key	Value
Scope	SIS
MessageType	Confirm
Sender	SISServer
Receiver	Remote

Step4. Register Components to SIS Server

Once each components compiled and is running on each terminal, they should be registered to SIS Server. This stage can be done through load each registration xml files of each component and send these messages to SIS Server. Load following files into PrjRemote and send each one to SIS server.

- CreateCancerDatasetDatabase.xml
- CreateCancerClassifierModule.xml
- CreatePCDCManager.xml
- CreateGeneOntology.xml


SIS Remote
Connection & Scope
Message(s)
Message To Send (Click Me To Save)
Message Received (Click Me To Save)
Send
Load
Clear

CreateCancerClassifierModule.xml
CreateCancerDatasetDatabase.xml
CreateGeneOntology.xml
CreatePCDCManager.xml

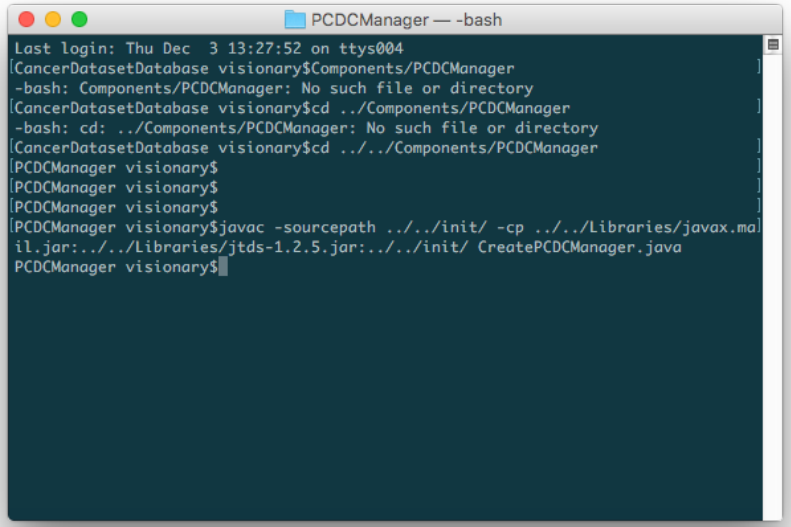
Key	Value
Scope	SIS.Scope1
MessageType	Register
Role	Basic
Name	CancerClassifierModule

Key	Value
Scope	SIS.Scope1
MessageType	Register
Role	Basic
Name	CancerClassifierModule

Step5. Compile Super Component
In our proposed prototype system, the super component is PCDC Manager. Once all other components are compiled and registered to SIS Server, now we can compile the supercomponent. The super component is located at 'Components/PCDCManager' folder and it can be compiled with following command.

- *javac -sourcepath ../../init/ -cp ../../Libraries/javax.mail.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreatePCDCManager.java*

Then, the screen will be look like following figure.

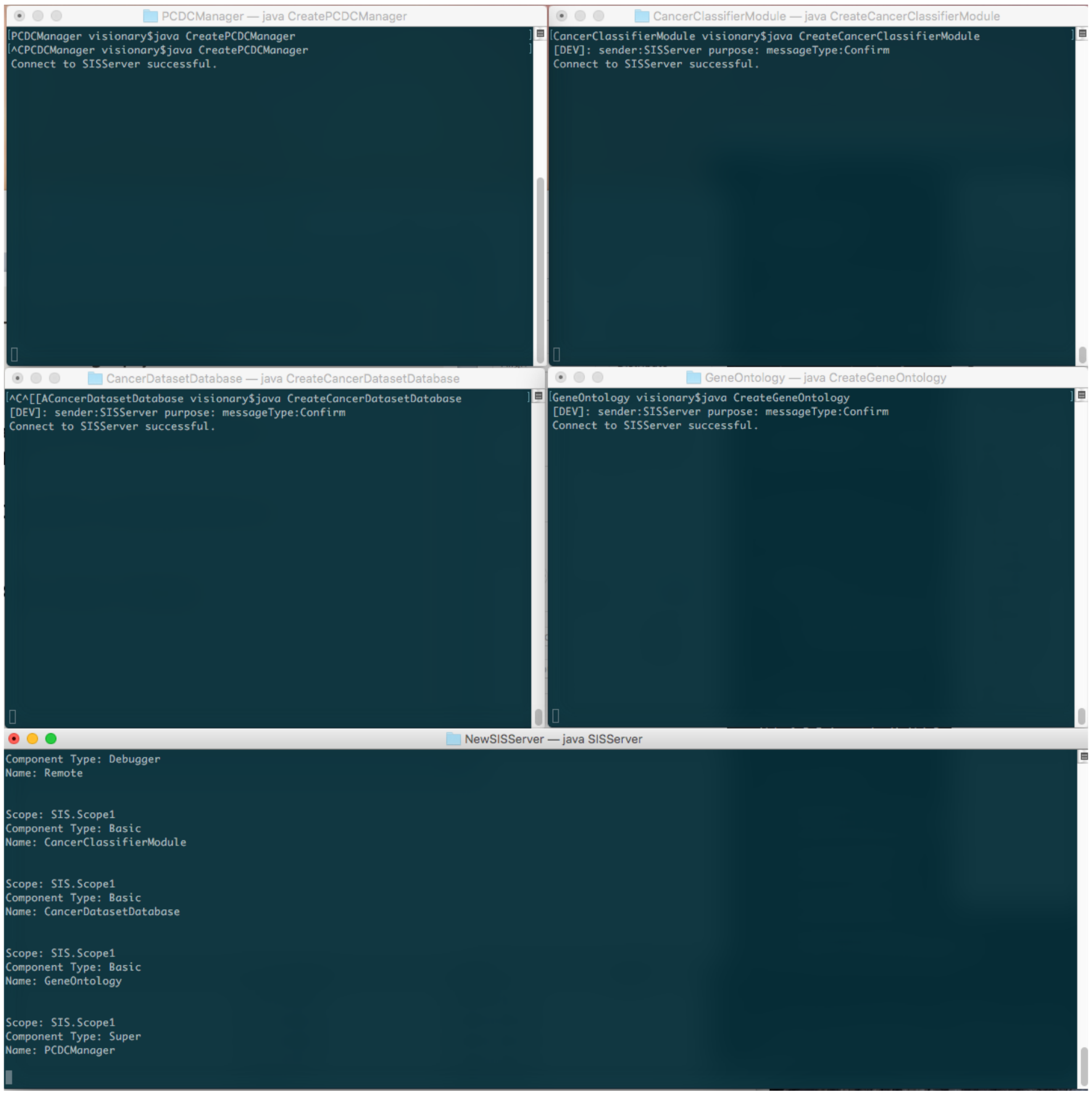


```
Last login: Thu Dec  3 13:27:52 on ttys004
CancerDatasetDatabase visionary$Components/PCDCManager
-bash: Components/PCDCManager: No such file or directory
CancerDatasetDatabase visionary$cd ../Components/PCDCManager
-bash: cd: ../Components/PCDCManager: No such file or directory
CancerDatasetDatabase visionary$cd ../../Components/PCDCManager
PCDCManager visionary$
PCDCManager visionary$
PCDCManager visionary$
PCDCManager visionary$javac -sourcepath ../../init/ -cp ../../Libraries/javax.ma
il.jar:../../Libraries/jtds-1.2.5.jar:../../init/ CreatePCDCManager.java
PCDCManager visionary$
```

Step6. Run all components
Now, all component can be operating on their command windows. The command for running all components is provided below.

- at Components/CancerClassifierModule
  java CreateCancerClassifierModule
- at Components/PCDCManager
  java CreatePCDCManager
- at Components/GeneOntology
  java CreateGeneOntology
- at Components/CancerDatasetDatabase
  java CreateCancerDatasetDatabase

Then, the each terminal windows will be look like following:


PCDCManager — java CreatePCDCManager
CancerClassifierModule — java CreateCancerClassifierModule
CancerDatasetDatabase — java CreateCancerDatasetDatabase
GeneOntology — java CreateGeneOntology
NewSISServer — java SISServer

```
PCDCManager visionary$java CreatePCDCManager
^CPCDCManager visionary$java CreatePCDCManager
Connect to SISServer successful.
```

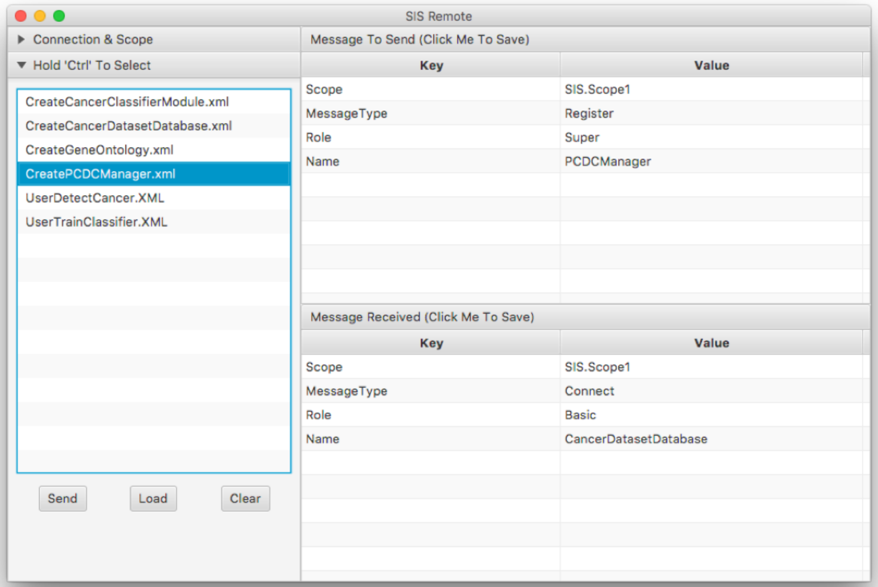
```
CancerClassifierModule visionary$java CreateCancerClassifierModule
[DEV]: sender:SISServer purpose: messageType:Confirm
Connect to SISServer successful.
```

```
^C^[[ACancerDatasetDatabase visionary$java CreateCancerDatasetDatabase
[DEV]: sender:SISServer purpose: messageType:Confirm
Connect to SISServer successful.
```

```
GeneOntology visionary$java CreateGeneOntology
[DEV]: sender:SISServer purpose: messageType:Confirm
Connect to SISServer successful.
```

```
Component Type: Debugger
Name: Remote


Scope: SIS.Scope1
Component Type: Basic
Name: CancerClassifierModule


Scope: SIS.Scope1
Component Type: Basic
Name: CancerDatasetDatabase


Scope: SIS.Scope1
Component Type: Basic
Name: GeneOntology


Scope: SIS.Scope1
Component Type: Super
Name: PCDCManager
```

Also, the PrjRemote will be looks like this; You must see the connected message on the 'Message Received' side of PrjRemote.



CreateCancerClassifierModule.xml
CreateCancerDatasetDatabase.xml
CreateGeneOntology.xml
CreatePCDCManager.xml
UserDetectCancer.XML
UserTrainClassifier.XML

Message To Send (Click Me To Save)

Key	Value
Scope	SIS.Scope1
MessageType	Register
Role	Super
Name	PCDCManager

Message Received (Click Me To Save)

Key	Value
Scope	SIS.Scope1
MessageType	Connect
Role	Basic
Name	CancerDatasetDatabase

At this moment, all preparation to run the prototype program is completed.

Step 7. Edit type of classifier and patient ID

Before running the program, you can configure type of classifier and patient ID through following steps.

- Open 'UserTrainCancer.XML' at xml/DataXML folder and edit *type of classifier*

  The 'UserTrainCancer.XML' file will be consisted of following elemetns..

```
<?xml version="1.0" standalone="yes"?>
<!--Generated by SISProjectCreator Version 1.0 -->
<Msg>
        <Item>
                <Key>Scope</Key>
                <Value>SIS.Scope1</Value>
        </Item>
        <Item>
                <Key>MessageType</Key>
                <Value>Alert</Value>
        </Item>
        <Item>
                <Key>Sender</Key>
                <Value>User</Value>
        </Item>
        <Item>
```

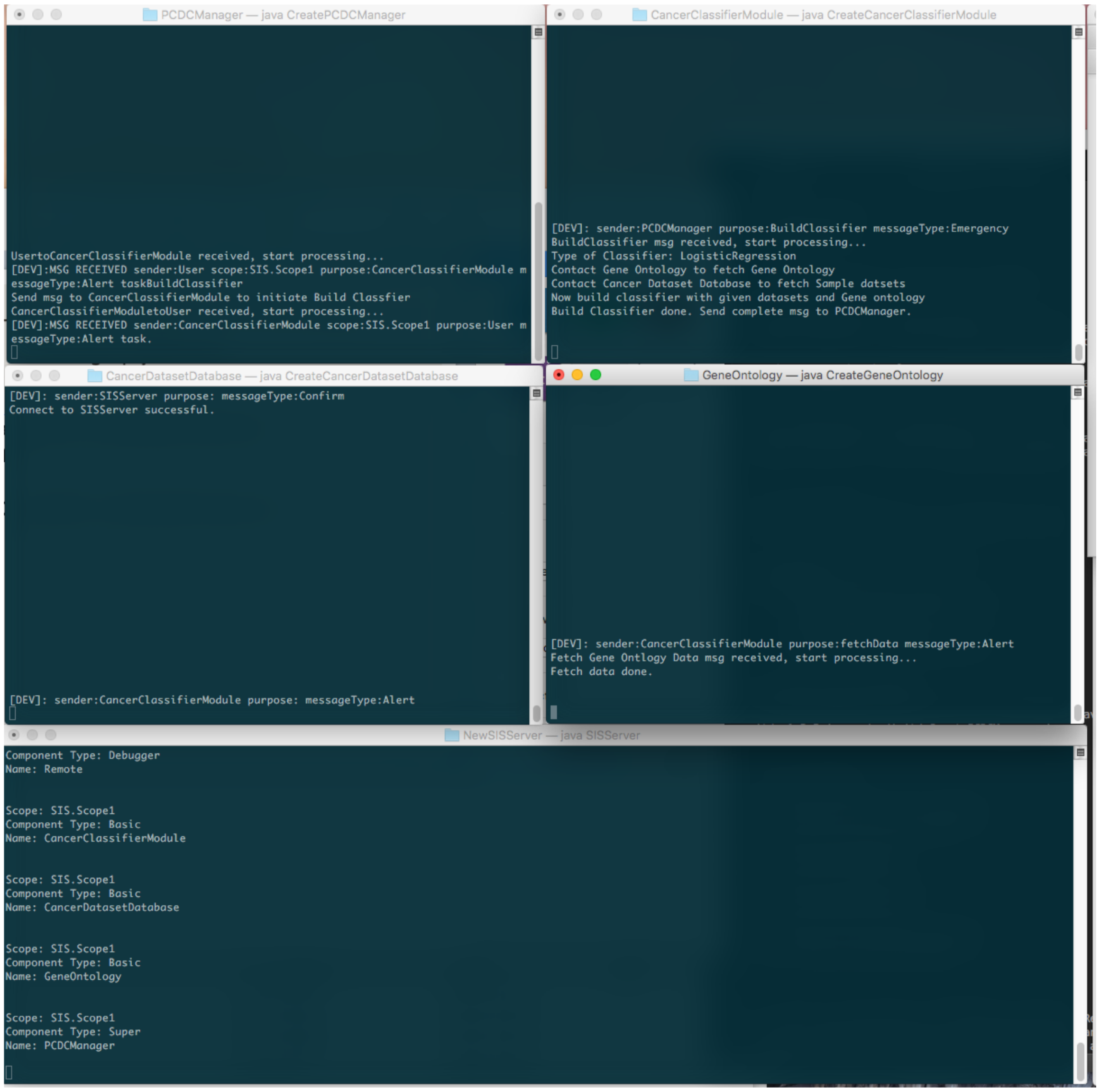
```
		<Key>Purpose</Key>
		<Value>CancerClassifierModule</Value>
	</Item>
	<Item>
		<Key>Task</Key>
		<Value>BuildClassifier</Value>
	</Item>
	<Item>
		<Key>classifierType</Key>
		<Value>LogisticRegression</Value>
	</Item>
</Msg>
```

- Open 'UserDetectCancer.XML' at xml/DataXML folder and edit *patient ID*
  The file will be consisted of like following.

```
<?xml version="1.0" standalone="yes"?>
<!--Generated by SISProjectCreator Version 1.0 -->
<Msg>
	<Item>
		<Key>Scope</Key>
		<Value>SIS.Scope1</Value>
	</Item>
	<Item>
		<Key>MessageType</Key>
		<Value>Alert</Value>
	</Item>
	<Item>
		<Key>Sender</Key>
		<Value>User</Value>
	</Item>
	<Item>
		<Key>Purpose</Key>
		<Value>CancerClassifierModule</Value>
	</Item>
	<Item>
		<Key>Task</Key>
		<Value>DetectCancer</Value>
	</Item>

	<Item>
		<Key>patientID</Key>
		<Value>10015</Value>
	</Item>
</Msg>
```

Step8. Train classifier

After editing configuration on step7, you can start classification through sending 'UserTrainCancer.XML' file at PrjRemote. Once this file is sent at PrjRemote, PCDC Manager will receive the message first and send classification task initiation message to Cancer Classifier Module (CCM). The CCM then send message to Cancer Dataset Database component and Gene Ontology component to fetch data. Please note that for this in-class project, since it is prototyping the overall system, no actual data is transmitted nor trained, but the all necessary sequence of messages and software logic for the task is designed and implemented for the prototype.


PCDCManager — java CreatePCDCManager
CancerClassifierModule — java CreateCancerClassifierModule
CancerDatasetDatabase — java CreateCancerDatasetDatabase
GeneOntology — java CreateGeneOntology
NewSISServer — java SISServer

```
UsertoCancerClassifierModule received, start processing...
[DEV]:MSG RECEIVED sender:User scope:SIS.Scope1 purpose:CancerClassifierModule m
essageType:Alert taskBuildClassifier
Send msg to CancerClassifierModule to initiate Build Classfier
CancerClassifierModuletoUser received, start processing...
[DEV]:MSG RECEIVED sender:CancerClassifierModule scope:SIS.Scope1 purpose:User m
essageType:Alert task.
```

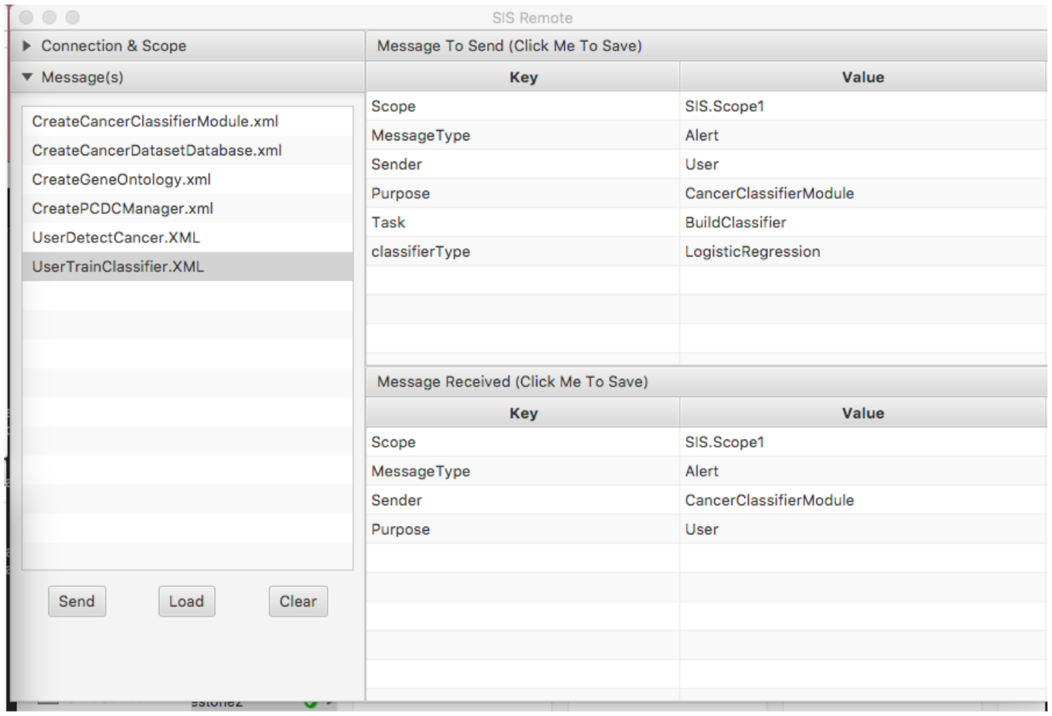
```
[DEV]: sender:PCDCManager purpose:BuildClassifier messageType:Emergency
BuildClassifier msg received, start processing...
Type of Classifier: LogisticRegression
Contact Gene Ontology to fetch Gene Ontology
Contact Cancer Dataset Database to fetch Sample datsets
Now build classifier with given datasets and Gene ontology
Build Classifier done. Send complete msg to PCDCManager.
```

```
[DEV]: sender:SISServer purpose: messageType:Confirm
Connect to SISServer successful.


[DEV]: sender:CancerClassifierModule purpose: messageType:Alert
```

```
[DEV]: sender:CancerClassifierModule purpose:fetchData messageType:Alert
Fetch Gene Ontlogy Data msg received, start processing...
Fetch data done.
```

```
Component Type: Debugger
Name: Remote


Scope: SIS.Scope1
Component Type: Basic
Name: CancerClassifierModule


Scope: SIS.Scope1
Component Type: Basic
Name: CancerDatasetDatabase


Scope: SIS.Scope1
Component Type: Basic
Name: GeneOntology


Scope: SIS.Scope1
Component Type: Super
Name: PCDCManager
```

Throughout this phase, a number of messages are send and received from component to component and once this phase is completed, user can see the final complete message from PCDC Manager at the PrjRemote.



CreateCancerClassifierModule.xml
CreateCancerDatasetDatabase.xml
CreateGeneOntology.xml
CreatePCDCManager.xml
UserDetectCancer.XML
UserTrainClassifier.XML

Message To Send (Click Me To Save)

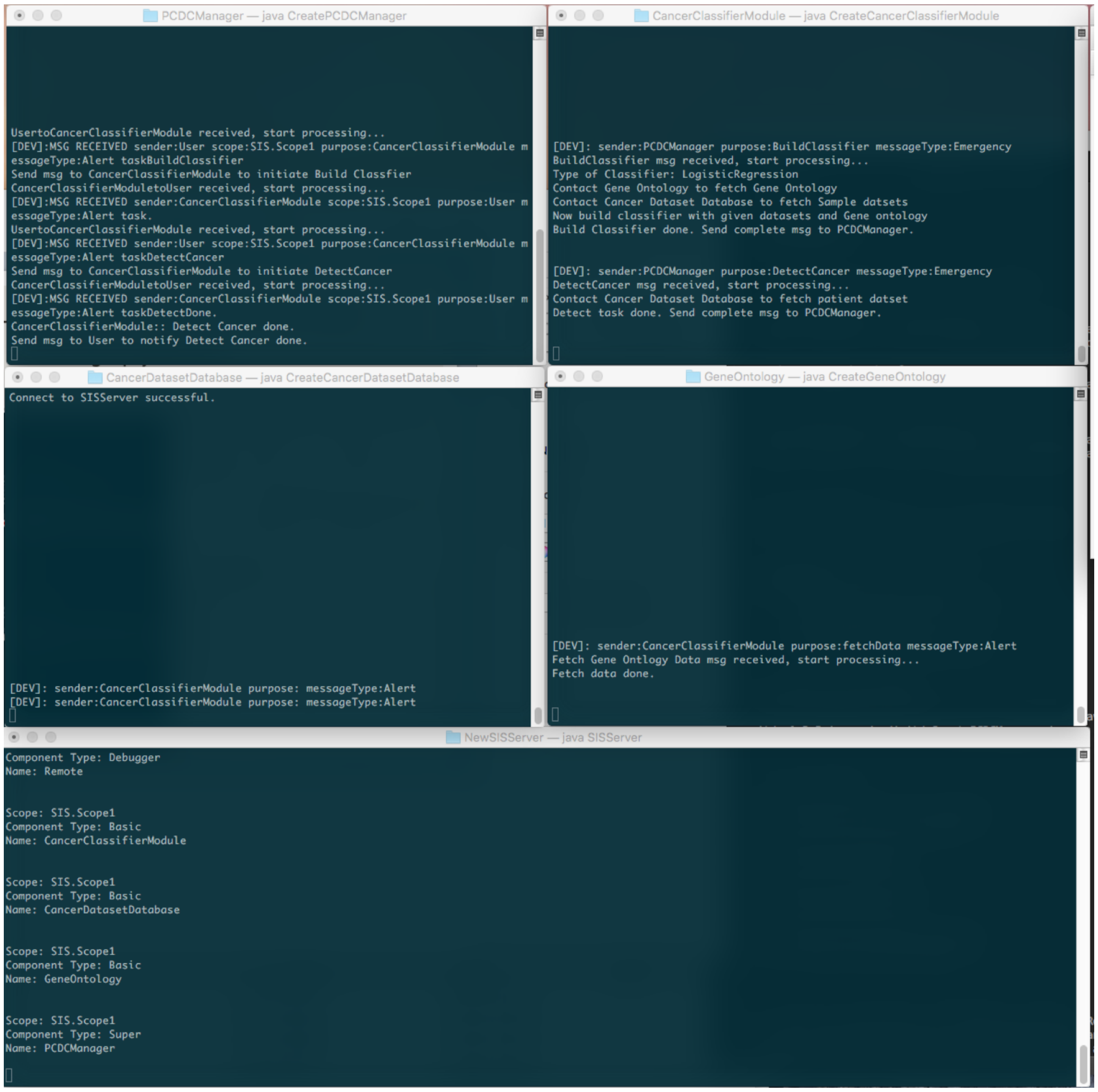
Key	Value
Scope	SIS.Scope1
MessageType	Alert
Sender	User
Purpose	CancerClassifierModule
Task	BuildClassifier
classifierType	LogisticRegression

Message Received (Click Me To Save)

Key	Value
Scope	SIS.Scope1
MessageType	Alert
Sender	CancerClassifierModule
Purpose	User

Step9. Detect Cancer

Likewise step8, once user configure patient ID on the 'UserTrainCancer.XML' file, user can initiated the detecting task by sending 'UserTrainCancer.XML' file to the PCDC Manager at the PrjRemote. Once this message is received at the PCDC Manager, the PCDC Manager will send task initiation message to the Cancer Classification Module (CCM) and then the CCM conducts task, and then it send back message to the PCDC Manager and the PCDC Manager send back to the user. Finally, user can see the complete message at the PrjRemote.



PCDCManager — java CreatePCDCManager

```
UsertoCancerClassifierModule received, start processing...
[DEV]:MSG RECEIVED sender:User scope:SIS.Scope1 purpose:CancerClassifierModule m
essageType:Alert taskBuildClassifier
Send msg to CancerClassifierModule to initiate Build Classfier
CancerClassifierModuletoUser received, start processing...
[DEV]:MSG RECEIVED sender:CancerClassifierModule scope:SIS.Scope1 purpose:User m
essageType:Alert task.
UsertoCancerClassifierModule received, start processing...
[DEV]:MSG RECEIVED sender:User scope:SIS.Scope1 purpose:CancerClassifierModule m
essageType:Alert taskDetectCancer
Send msg to CancerClassifierModule to initiate DetectCancer
CancerClassifierModuletoUser received, start processing...
[DEV]:MSG RECEIVED sender:CancerClassifierModule scope:SIS.Scope1 purpose:User m
essageType:Alert taskDetectDone.
CancerClassifierModule:: Detect Cancer done.
Send msg to User to notify Detect Cancer done.
```

CancerClassifierModule — java CreateCancerClassifierModule

```
[DEV]: sender:PCDCManager purpose:BuildClassifier messageType:Emergency
BuildClassifier msg received, start processing...
Type of Classifier: LogisticRegression
Contact Gene Ontology to fetch Gene Ontology
Contact Cancer Dataset Database to fetch Sample datsets
Now build classifier with given datasets and Gene ontology
Build Classifier done. Send complete msg to PCDCManager.


[DEV]: sender:PCDCManager purpose:DetectCancer messageType:Emergency
DetectCancer msg received, start processing...
Contact Cancer Dataset Database to fetch patient datset
Detect task done. Send complete msg to PCDCManager.
```

CancerDatasetDatabase — java CreateCancerDatasetDatabase

```
Connect to SISServer successful.

[DEV]: sender:CancerClassifierModule purpose: messageType:Alert
[DEV]: sender:CancerClassifierModule purpose: messageType:Alert
```

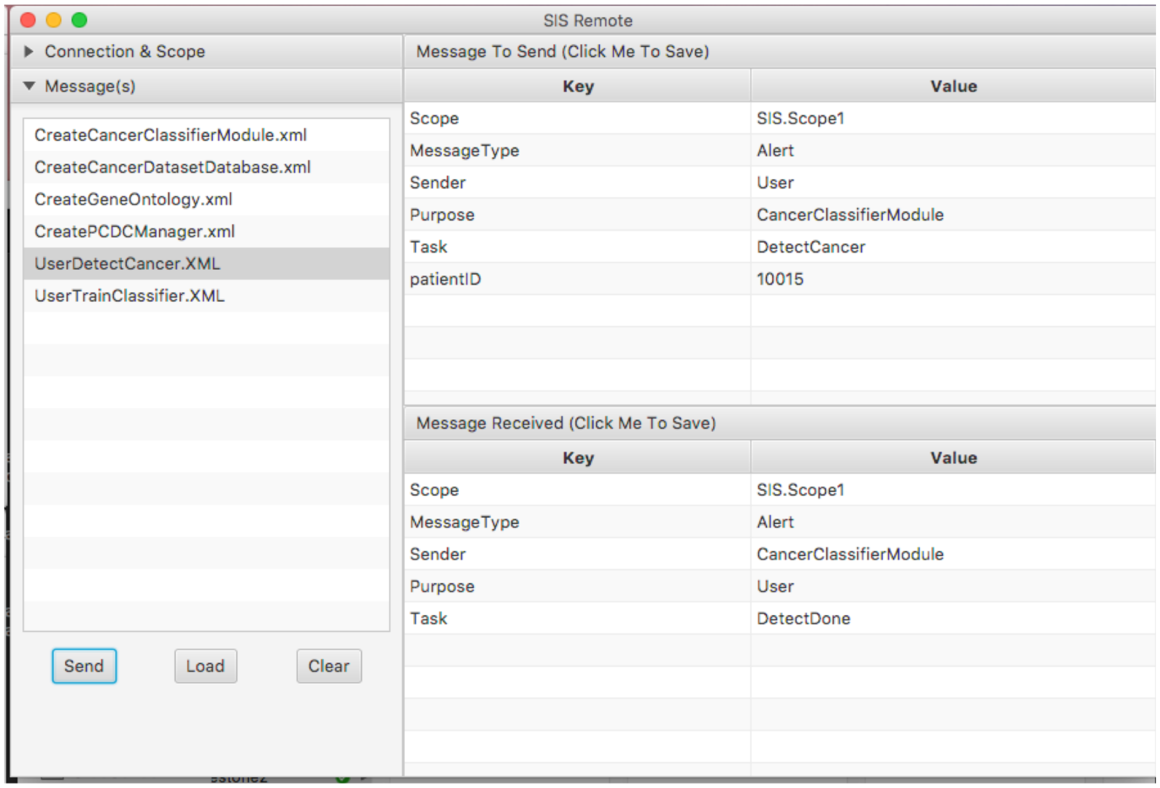
GeneOntology — java CreateGeneOntology

```
[DEV]: sender:CancerClassifierModule purpose:fetchData messageType:Alert
Fetch Gene Ontlogy Data msg received, start processing...
Fetch data done.
```

NewSISServer — java SISServer

```
Component Type: Debugger
Name: Remote


Scope: SIS.Scope1
Component Type: Basic
Name: CancerClassifierModule


Scope: SIS.Scope1
Component Type: Basic
Name: CancerDatasetDatabase


Scope: SIS.Scope1
Component Type: Basic
Name: GeneOntology


Scope: SIS.Scope1
Component Type: Super
Name: PCDCManager
```

The final message can be seen at PrjRemote is look like following;



CreateCancerClassifierModule.xml
CreateCancerDatasetDatabase.xml
CreateGeneOntology.xml
CreatePCDCManager.xml
UserDetectCancer.XML
UserTrainClassifier.XML

Message To Send (Click Me To Save)

Key	Value
Scope	SIS.Scope1
MessageType	Alert
Sender	User
Purpose	CancerClassifierModule
Task	DetectCancer
patientID	10015

Message Received (Click Me To Save)

Key	Value
Scope	SIS.Scope1
MessageType	Alert
Sender	CancerClassifierModule
Purpose	User
Task	DetectDone

6. Conclusion

New prototype that cancer classification and detection based on patient's gene dataset within Slow Intelligent System framework is shown. Especially, successful building of each component of the system and communication and between the components were demonstrated. It is expected that this approach improves the quality of life of potential users of the personalized healthcare system in Slow Intelligent System. In future, further implementation coupling this with existing machine learning classification framework is needed.