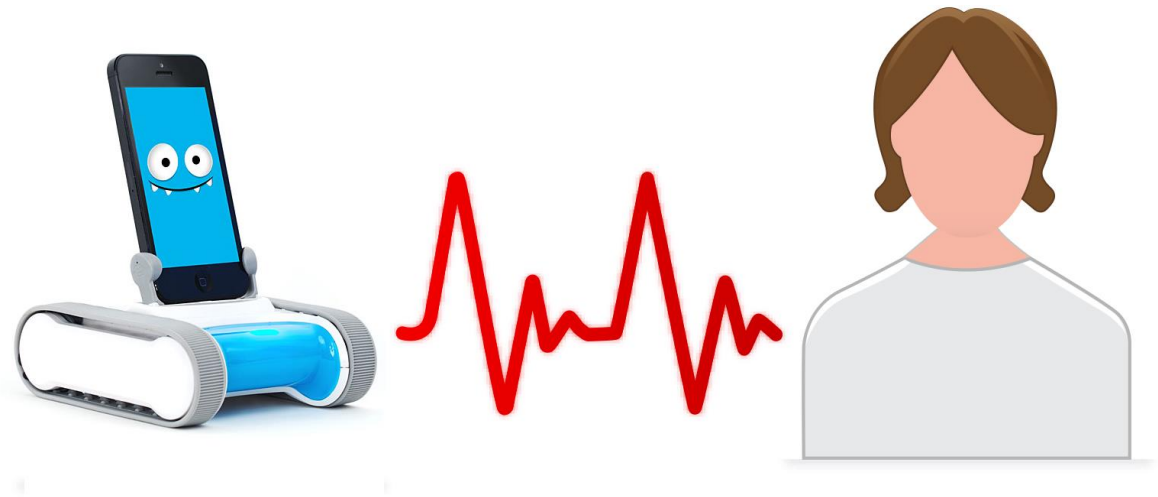
iROBO

■ *Patient's Personal Assistant*



Contents

1. PROPOSAL

1.1. Motivation:

In medical field in any hospital patients feel so lonely after some days and even research proved that constant attention helps healing faster than keeping them lonely. So this robot will interact with patients based on their mood swing. This will even help doctors to keep on monitoring patient under doctor's radar.

1.2. Significance/Uniqueness:

Say that patient is happy, it will ask in cheering voice that " How are you sir, Seems like you are so happy, is it because of any happy memories, or any good news, can you please share with me too…" or if patient is so sad, based on his medical history it will talk saying that " You seem so sad sir, don't feel that way, be happy, Can I sing some songs, can I dance <to make it laugh>, can I tell joke"

1.3. Objectives:

This will help patient think in good mood and won't make them feel lonely. Also in hospital room it will keep on monitoring the monitors and will keep on updating doctors on the status of the patient by using patient id and name using messaging/email. Robot might even suggest calling particular people based on mood swings. This robot might even translate from one language to another where patient will talk in one language and it talks back to doctor in another language.

1.4. System Features:

Features May Include:

- Constant Relay of Front Cam video to Source Phone
- Movement by speech
- Image Recognition/Mood Recognition
- Object Recognition
- Patient Records search
- Language Translation

This Robot will also assist the people who can't speak/hear by identifying sign language and showing in form of text on screen on what it want to say.

Furthermore scope can be expanded based on the user's work/needs.

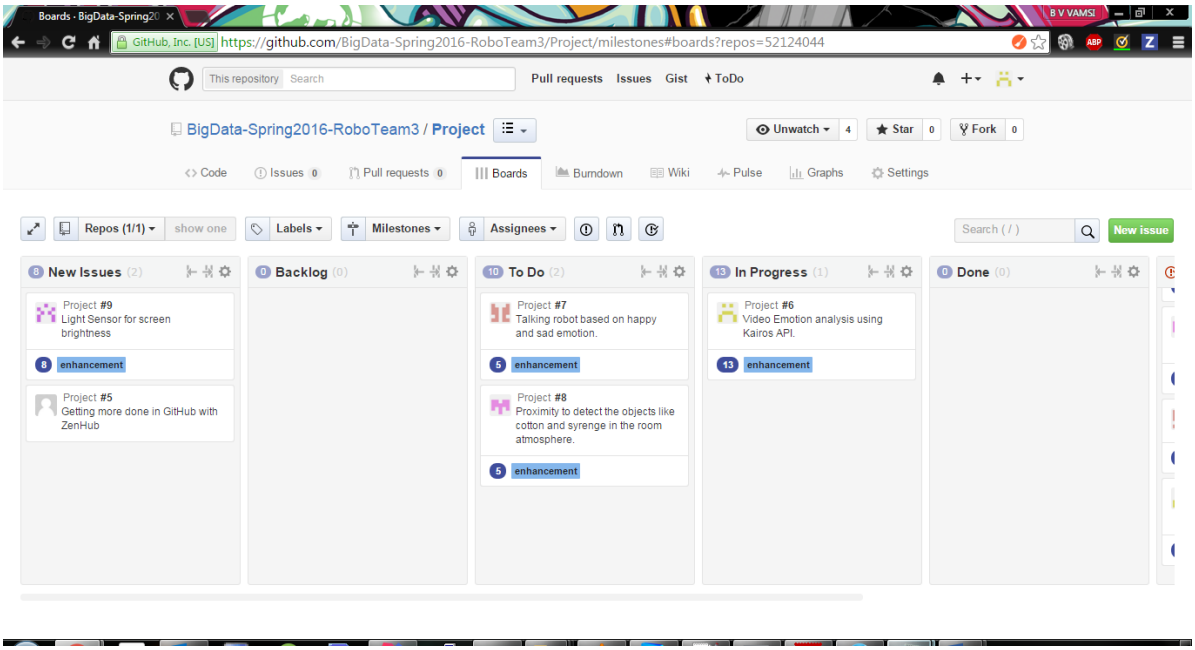
1.5. Related Work:

Till now some specialized hospitals have robots which will just assist doctors while attending patients and nowhere personalized robots are manufactured/coded.

2. Project Plan

2.1. Features Designed & Implemented:

- Identifying the emotion analysis in video, but currently we are able to identify the face in an image, this video analysis is scheduled for next increment. (faceplusplus API, kairos API).
- Proximity to detect the objects.
- Light sensor used to detect the atmosphere lightings.
- Getting the Heartbeat as notification to Smartwatch based on the critical changes to heart beat.



2.2. Project Timelines, Members, Task Responsibility

2.2.1. Timelines:

Date	Increment #
2/19/2016	Plan & Project Increment 1
3/11/2016	Project Increment 2
4/6/2016	Project Increment 3
4/29/2016	Project Increment 4

2.2.2. Members:

A) VENKATA VAMSI KRISHNA BHUVANAM
B) SOWMYA YELMATI
C) VARUN CHAVAKULA
D) VIKAS KONDAPALLI

2.2.3. Task Responsibility:

A) SPARK Machine Learning Tasks: Vikas Kondapalli
B) API and Emotion analysis: Venkata Vamsi Krishna Bhuvanam
C) ROBO Movements & Speech : Sowmya Yelmati
D) Speech emotion Analysis: Varun Chavakula

3. First Increment Report

3.1. Existing API:

- We will be using Kairos Emotion Analysis API for analyzing the patient's face in the video and based on the emotion, particular action will be taken.
- Also we will be using Speech to Text API so that the robo will interact with patient based on the training set provided to it.

3.2. Design of Features:

3.2.1. Hadoop/Machine Learning Algorithms:

3.2.1.1. Supervised Learning:

For Supervised Learning we are planning to use K- Nearest Neighbors or Decision Tree Machine Learning Algorithm.

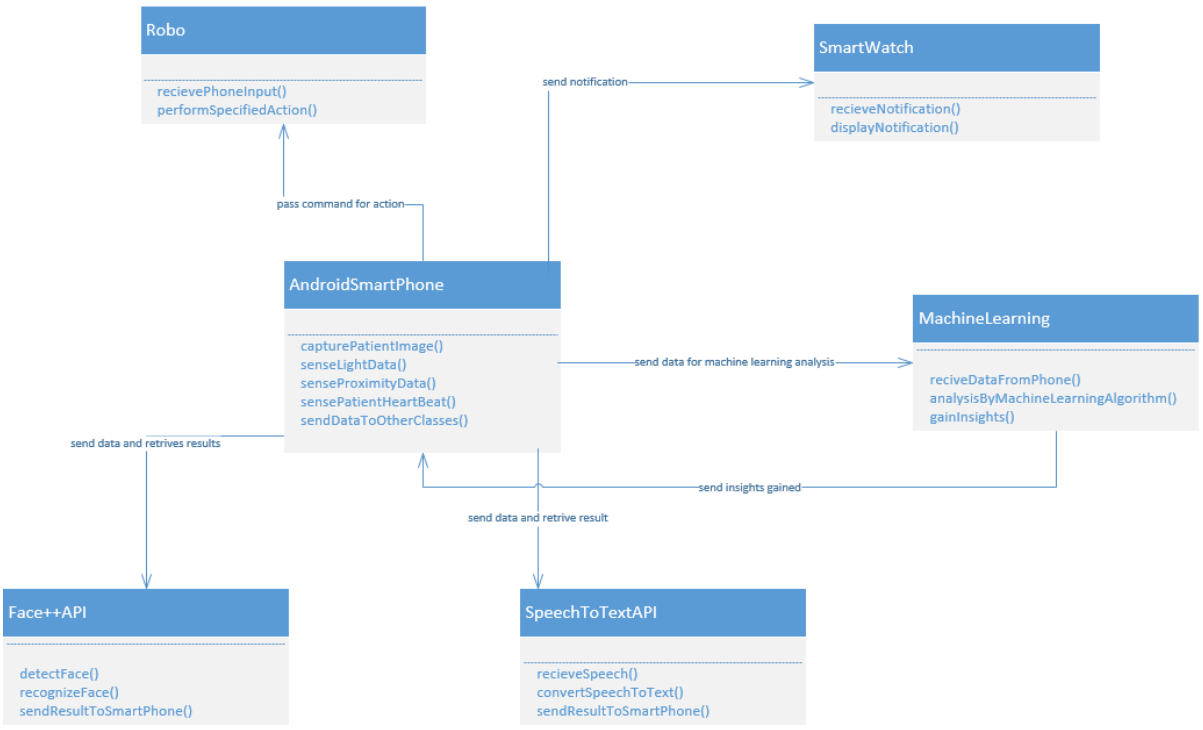
Steps in Supervised Learning:

- Prepare Data
- Choose an Algorithm
- Fit a Model
- Choose a Validation Method
- Examine Fit and Update Until Satisfied
- Use Fitted Model for Predictions

3.2.1.2. Un-Supervised Learning:

- For un-supervised Learning we are planning to use K- Means Clustering Algorithm.

3.2.2. Class Diagram:



This class diagram is only up to the design and features which have been planned till now, this design may modify and get completed in future increments.

3.2.2.1. Explanation:

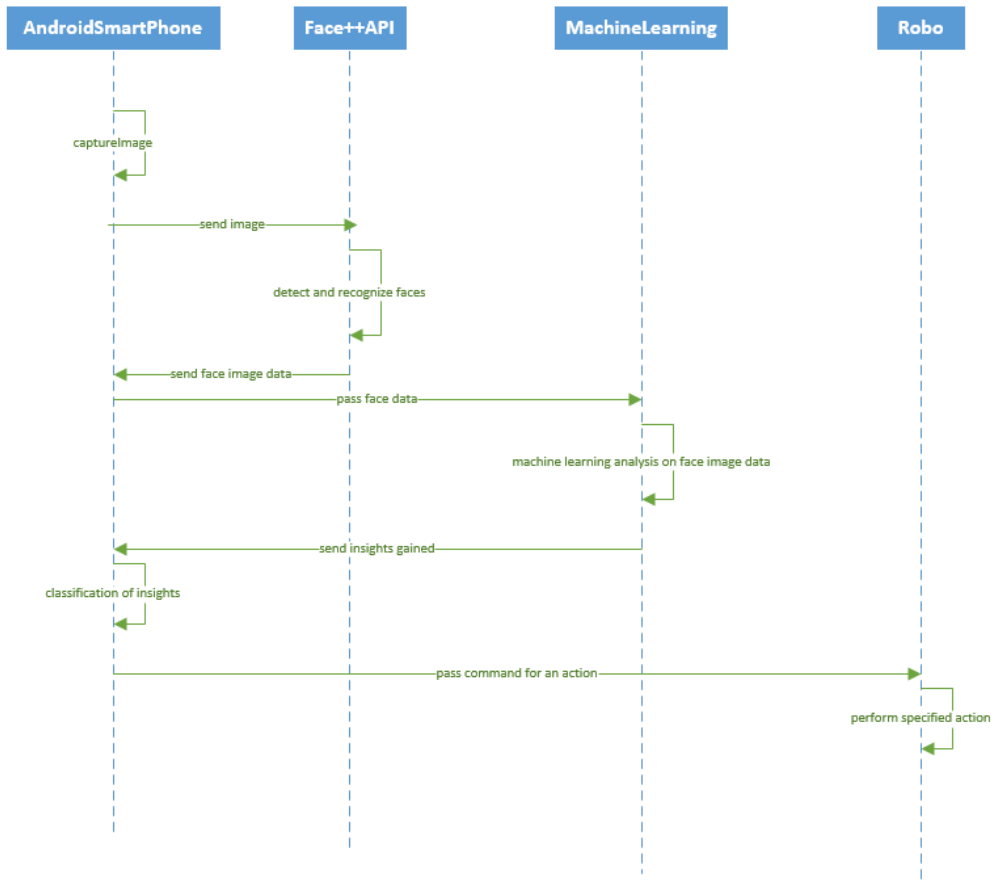
The classes in this design are

- Robo: This class represents the actual Robo which is used in the project. This class receives the command from smartphone and performs the specified action. The actions include movement action, talking action etc.
- Smartwatch: This class represents the actual android smartwear which is used in the project. This class receives the notification from the smart phone and displays the corresponding notification.
- Face++ API: This class represents the open source Face++ API. It receives the image data from the smartphone and detects, recognizes the faces present in the image.
- SpeechToTextAPI: This class represents the open source Speech to text API which is used in this project. This class receives the audio data from smart phone and converts the speech in it to text.
- MachineLearning: This class represents the Machine Learning module in this project. This class receives data from smart phone and machine learning analysis on the data in Spark using Spark machine learning library to recognize the patterns and gain meaningful insights.

- AndroidSmartPhone: This is the center class of the design which represents the android phone used in the project. This class collects the data from different sources. It send the data like image and speech to Face++ and SpeechToText APIs respectively. It then sends this data to machine Learning module for analysis. It then receives results from machine learning module and sends notifications to smartwatch and commands to Robo based upon the context.

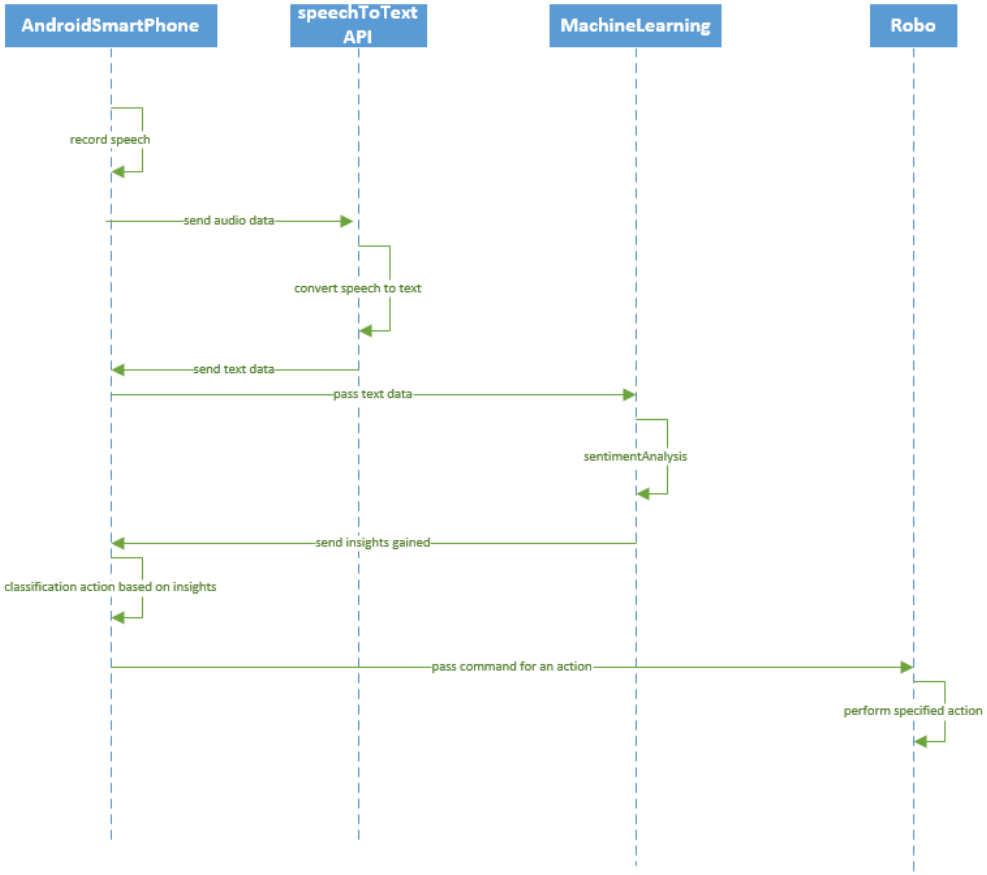
3.2.3. Sequence Diagram:

Sequence diagram for emotion analysis through patient face:



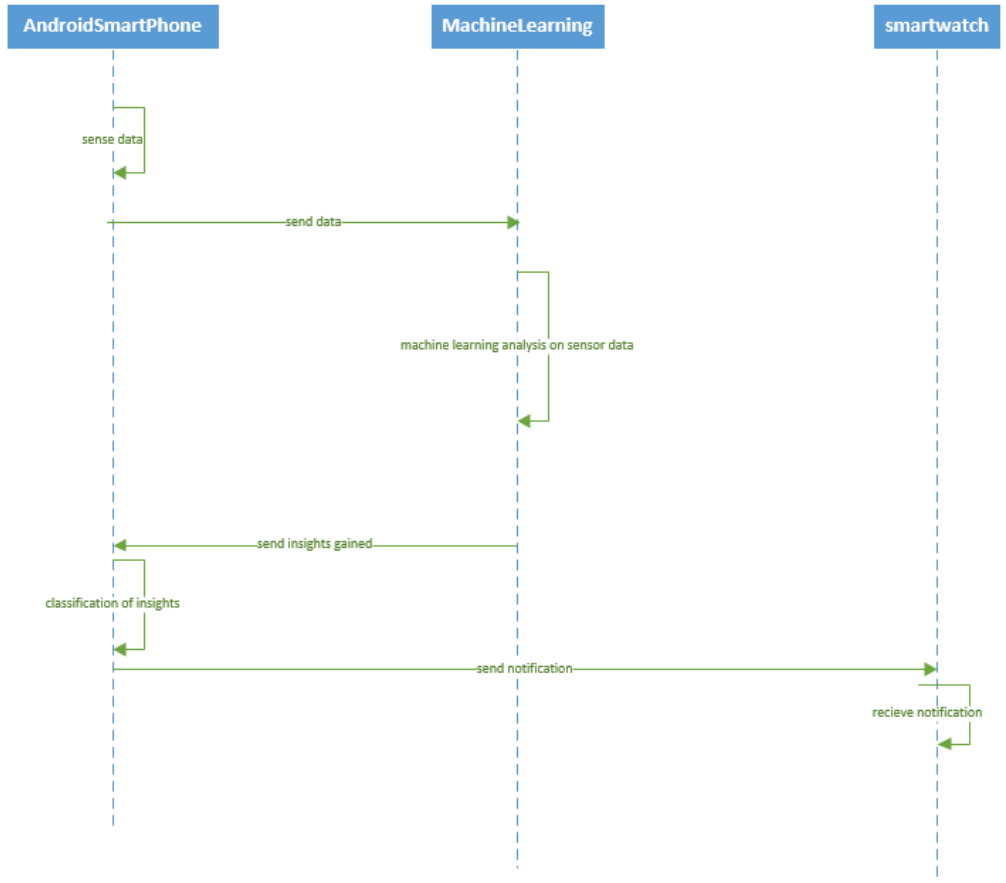
- In this Sequence Diagram, Android smartphone captures patient images and sends that data to Face++ API.
- Face++ API then processes that image and detects the face part of the image and sends that data to mobile client.
- The mobile client sends this data to machine learning module in data is analyzed to gain useful insights and pass that data to mobile client. This insights are classified and corresponding commands are passed to Robo.

3.2.4. Sequence Diagram for emotion analysis through patient speech:



- In this Sequence Diagram, Android smartphone records patient speech and sends that data to Speech to Text API.
- Speech To Text API then processes that audio data and converts it into text.
- The mobile client sends this data to machine learning module in data is analyzed to gain useful insights and pass that data to mobile client. This insights are classified and corresponding commands are passed to Robo.

3.2.5. Sequence diagram for detecting environmental factors:

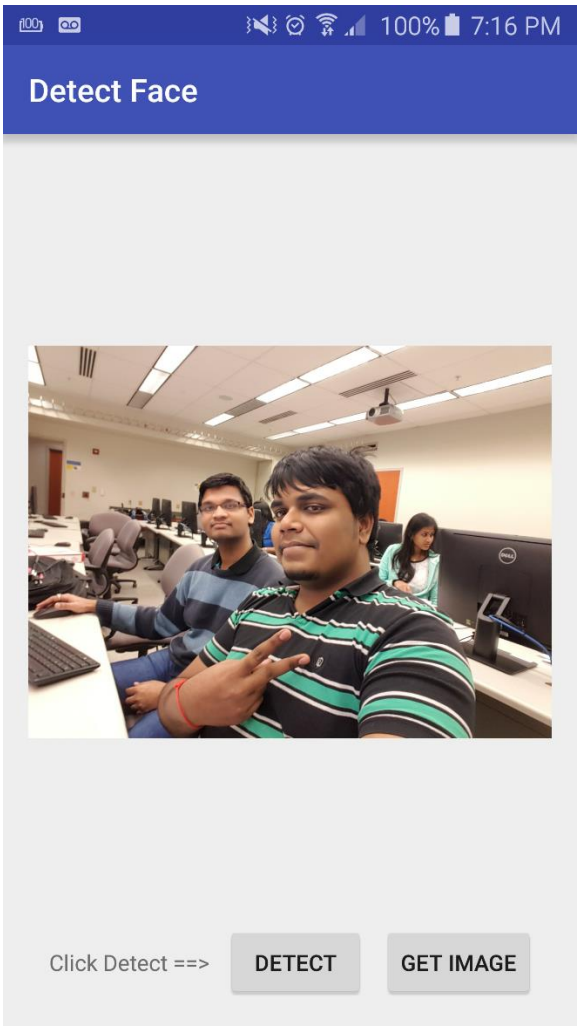


- In this Sequence Diagram, Android smartphone senses the environmental data such as Lighting, any nearby objects and patient heartbeat.
- The mobile client sends this data to machine learning module in data is analyzed to gain useful insights and pass that data to mobile client. This insights are classified and corresponding notifications are passed to Smart watch.

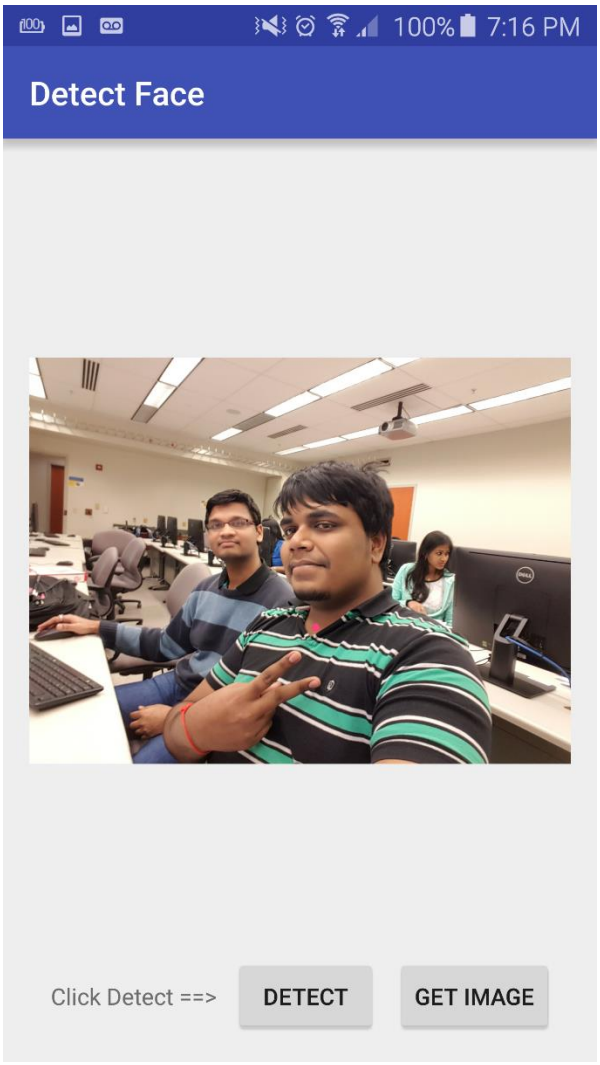
3.3. Implementation:

3.3.1. Face Detection:

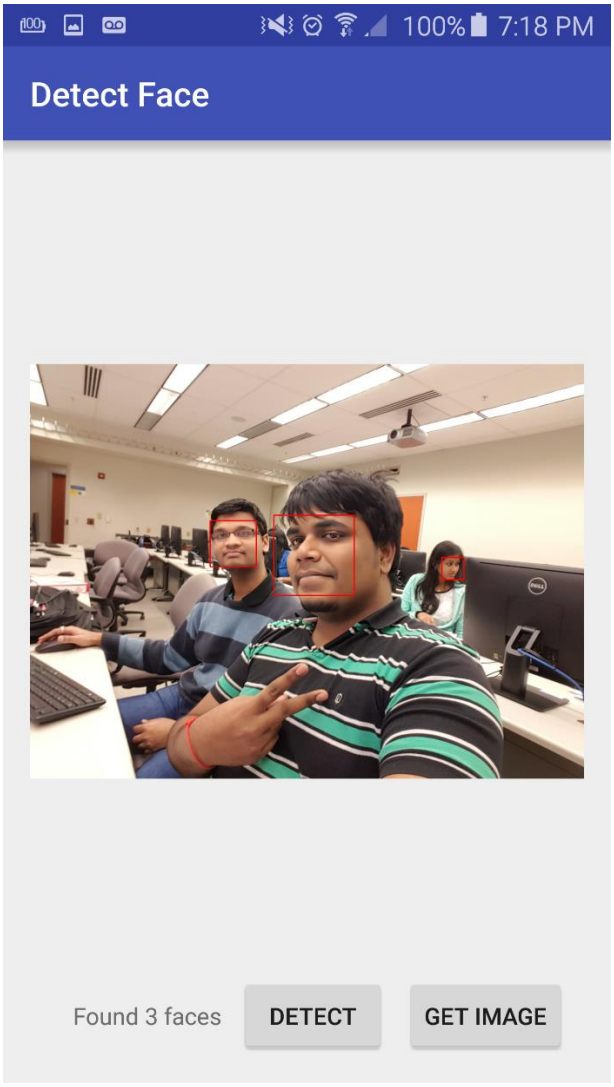
➔ First select the image from the gallery



➔ Click on Detect Button (which will be triggered automatically on the camera active method)

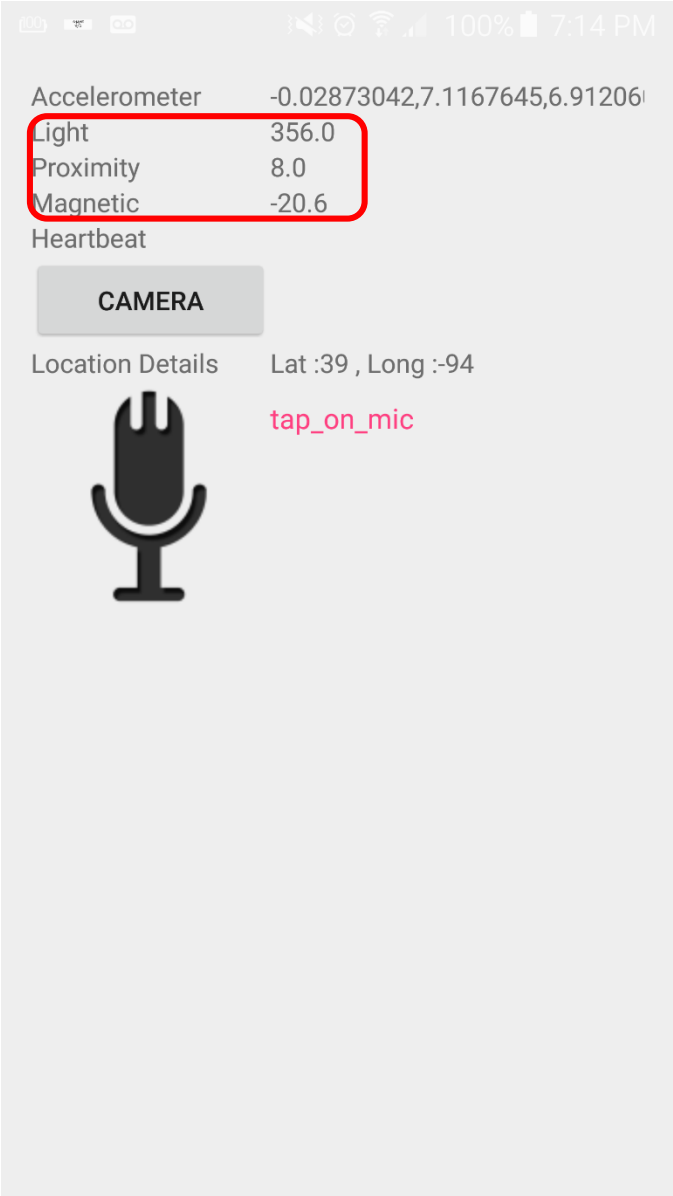


➔ Total faces in the image will be detected and highlighted. (In the same way using Kairos api we will detect the emotion in a video)

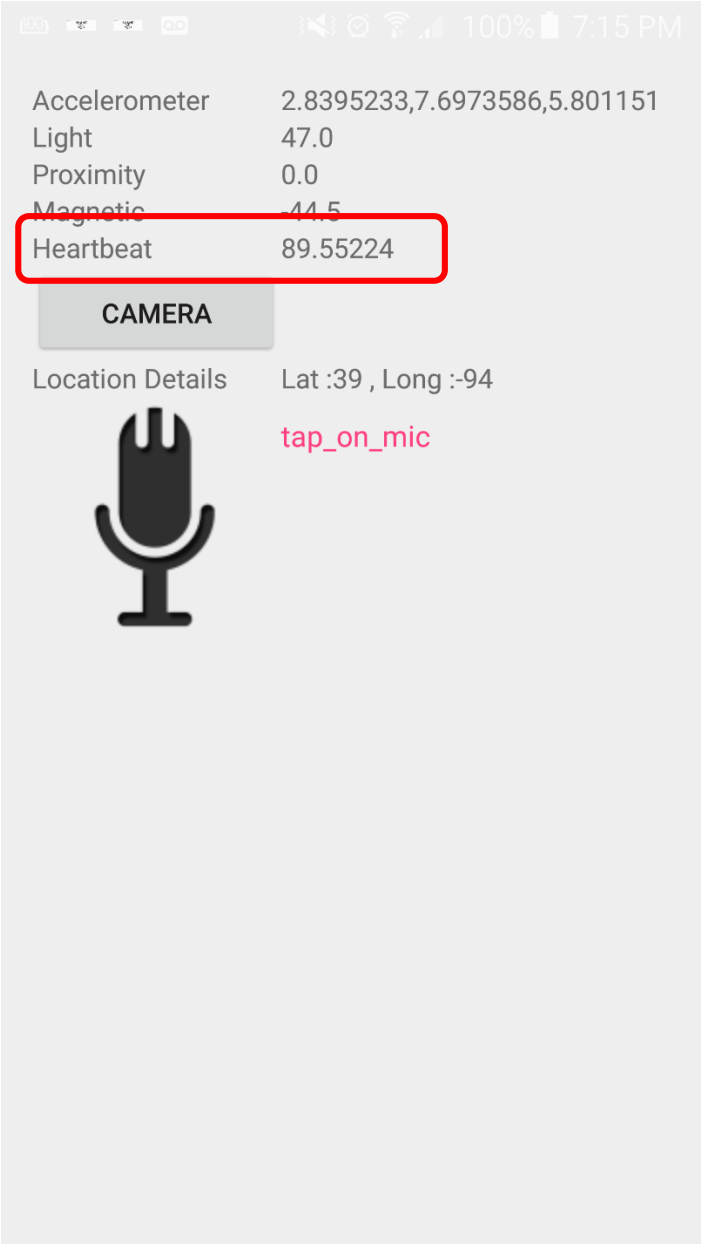


3.3.2. Light Sensor and Heart Beat & Proximity Sensor:

➔ Light and Proximity Sensors will be sensed and the proper action will be performed by robo.



➔ Heart-beat of the patient will be recorded and the prompt action like sending alert to doctors if fallen/risen of heart beat to critical stage.



3.4. Deployment:

Github for Face Plus Plus: https://github.com/BigDataSP2016TutorialLab/Lab-1/tree/master/Lab-4/Source

Github for Light Sensor and Proximity Sensor: https://github.com/Vikas-kondapalli/Lab_Assignments/tree/master/source/Assignment3_sourcecode/Lab3

4. Second Increment Implementation:

4.1. Mobile Phone to Spark Connection:

Here we have created an android application which will send the data stream to spark by creating a socket stream connection. We have implemented speech to Text implementation and sending that text to Spark Server.

4.2. Sending data to Spark:

Here the data is sent to Spark server by creating a socket streaming connection with spark and sending data over to Spark.

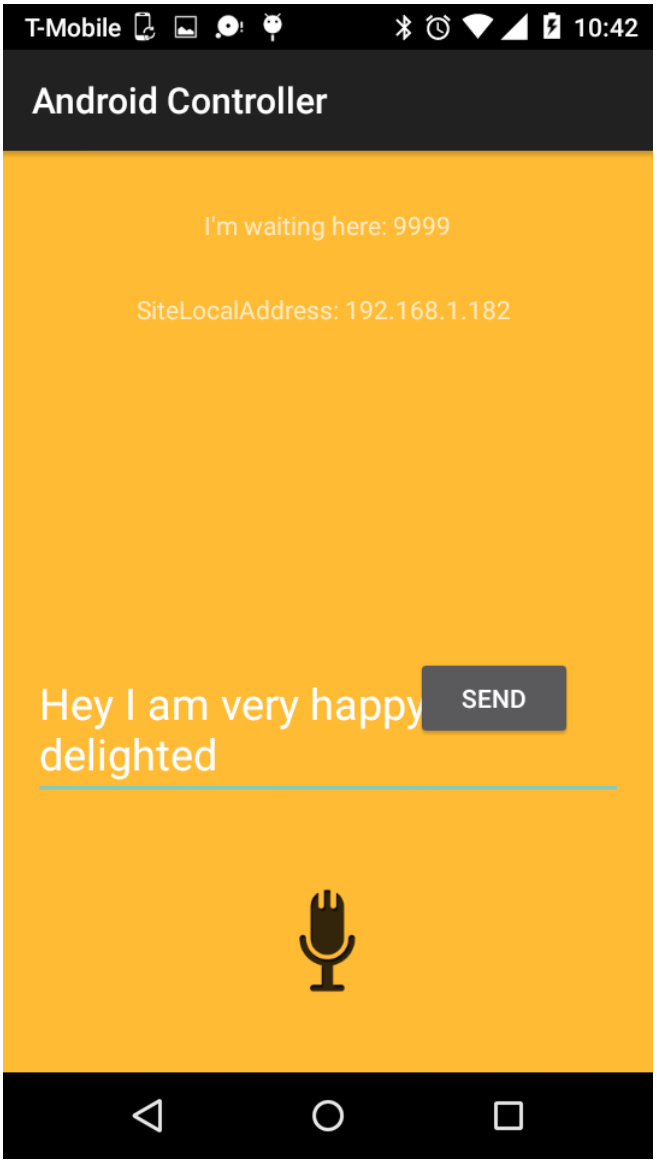
4.3. Sentiment Analysis on Spark:

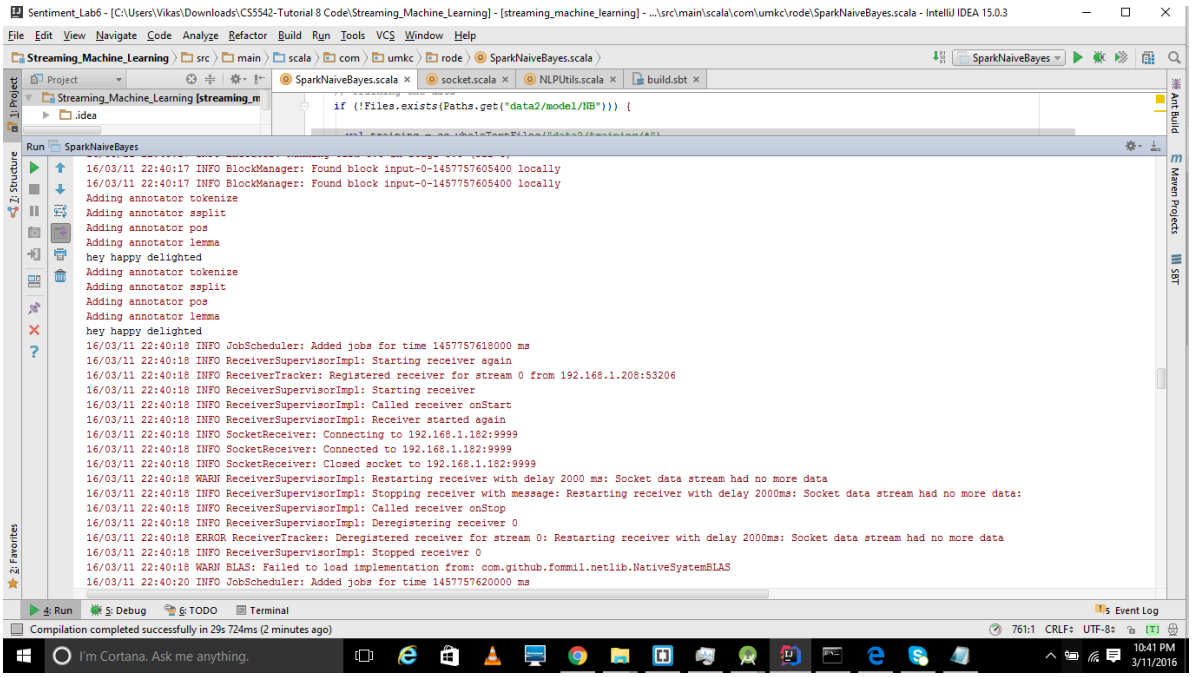
Here once the data was sent to Spark server, that data will be analyzed using Stanford NLP and predicted whether the sent data is positive or negative scenario related.

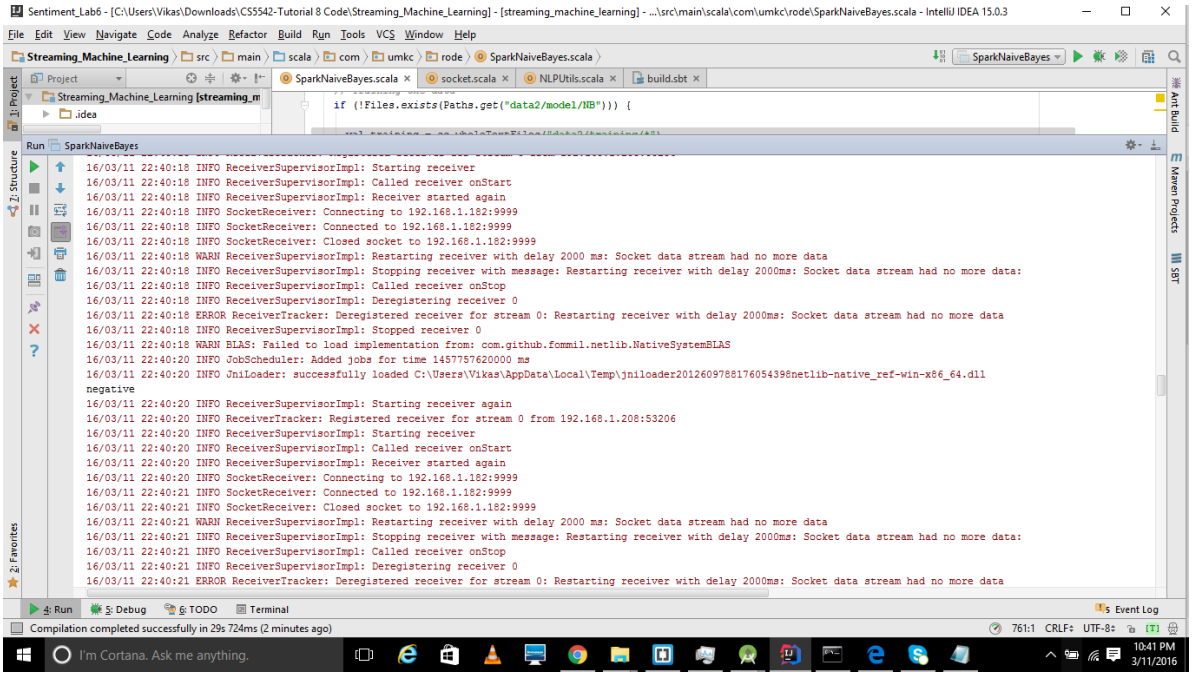
4.4. Getting Response back to Phone from Spark:

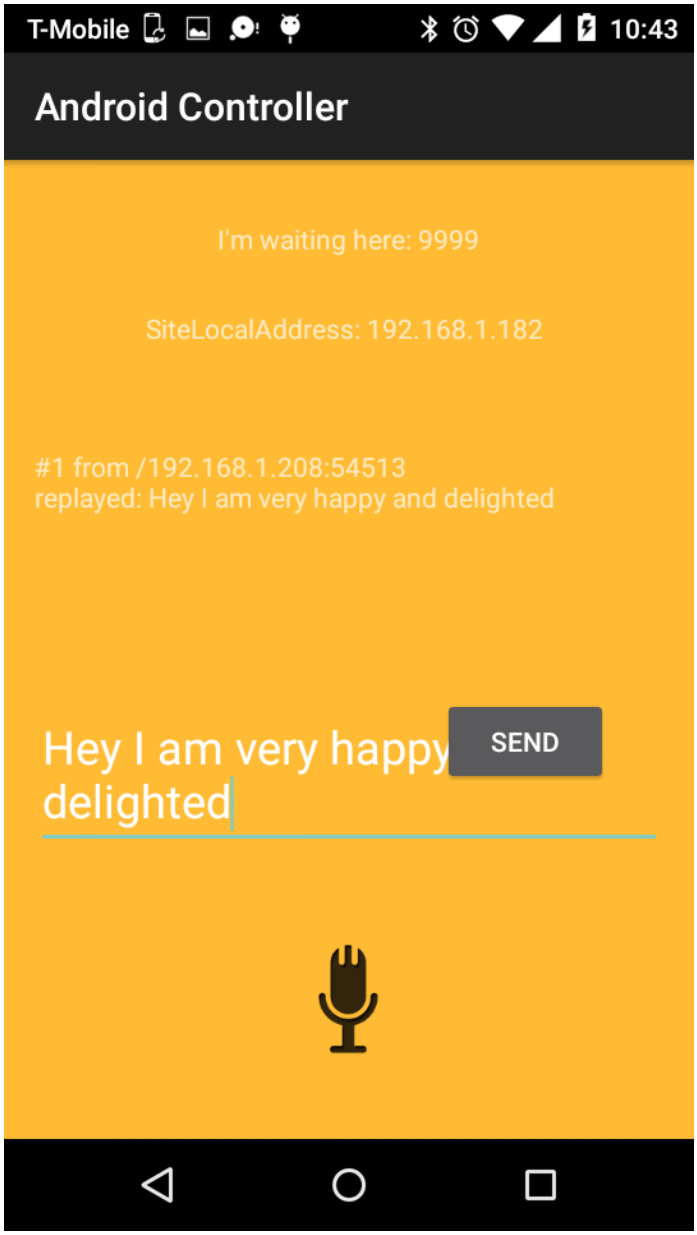
Based on the information analyzed by Stanford NLP the analyzed data will be sent back to phone where the android phone will do further analysis on that and will perform particular set of operations based on output. For example if output is positive, then robot/android phone will ask “You seem to be in happy/nice mood. How are you doing today?” If the output is negative, then the robot/android phone will ask “You seem so sad/feeling down today. Feel better soon. Can I play a song for you or tell a joke”

5. Second Increment Screenshots:




Sentiment_Lab6 - [C:\Users\Vikas\Downloads\CS5542-Tutorial 8 Code\Streaming_Machine_Learning] - [streaming_machine_learning] - ...\src\main\scala\com\umkc\rode\SparkNaiveBayes.scala - IntelliJ IDEA 15.0.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Streaming_Machine_Learning src main scala com umkc rode SparkNaiveBayes.scala
SparkNaiveBayes
SparkNaiveBayes.scala socket.scala NLPUtils.scala build.sbt
Project
Streaming_Machine_Learning [streaming_m
.idea
if (!Files.exists(Paths.get("data2/model/NB"))) {
Run SparkNaiveBayes
16/03/11 22:40:17 INFO BlockManager: Found block input-0-1457757605400 locally
16/03/11 22:40:17 INFO BlockManager: Found block input-0-1457757605400 locally
Adding annotator tokenize
Adding annotator ssplit
Adding annotator pos
Adding annotator lemma
hey happy delighted
Adding annotator tokenize
Adding annotator ssplit
Adding annotator pos
Adding annotator lemma
hey happy delighted
16/03/11 22:40:18 INFO JobScheduler: Added jobs for time 1457757618000 ms
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Starting receiver again
16/03/11 22:40:18 INFO ReceiverTracker: Registered receiver for stream 0 from 192.168.1.208:53206
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Starting receiver
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Called receiver onStart
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Receiver started again
16/03/11 22:40:18 INFO SocketReceiver: Connecting to 192.168.1.182:9999
16/03/11 22:40:18 INFO SocketReceiver: Connected to 192.168.1.182:9999
16/03/11 22:40:18 INFO SocketReceiver: Closed socket to 192.168.1.182:9999
16/03/11 22:40:18 WARN ReceiverSupervisorImpl: Restarting receiver with delay 2000 ms: Socket data stream had no more data
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Stopping receiver with message: Restarting receiver with delay 2000ms: Socket data stream had no more data:
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Called receiver onStop
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Deregistering receiver 0
16/03/11 22:40:18 ERROR ReceiverTracker: Deregistered receiver for stream 0: Restarting receiver with delay 2000ms: Socket data stream had no more data
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Stopped receiver 0
16/03/11 22:40:18 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
16/03/11 22:40:20 INFO JobScheduler: Added jobs for time 1457757620000 ms
4: Run 5: Debug 6: TODO Terminal Event Log
Compilation completed successfully in 29s 724ms (2 minutes ago)
761:1 CRLF: UTF-8:
I'm Cortana. Ask me anything.
10:41 PM
3/11/2016


Sentiment_Lab6 - [C:\Users\Vikas\Downloads\CS5542-Tutorial 8 Code\Streaming_Machine_Learning] - [streaming_machine_learning] - ...\src\main\scala\com\umkc\rode\SparkNaiveBayes.scala - IntelliJ IDEA 15.0.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Streaming_Machine_Learning src main scala com umkc rode SparkNaiveBayes.scala
SparkNaiveBayes
SparkNaiveBayes.scala socket.scala NLPUtils.scala build.sbt
Project
Streaming_Machine_Learning [streaming_m
.idea
if (!Files.exists(Paths.get("data2/model/NB"))) {
Run SparkNaiveBayes
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Starting receiver
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Called receiver onStart
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Receiver started again
16/03/11 22:40:18 INFO SocketReceiver: Connecting to 192.168.1.182:9999
16/03/11 22:40:18 INFO SocketReceiver: Connected to 192.168.1.182:9999
16/03/11 22:40:18 INFO SocketReceiver: Closed socket to 192.168.1.182:9999
16/03/11 22:40:18 WARN ReceiverSupervisorImpl: Restarting receiver with delay 2000 ms: Socket data stream had no more data
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Stopping receiver with message: Restarting receiver with delay 2000ms: Socket data stream had no more data:
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Called receiver onStop
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Deregistering receiver 0
16/03/11 22:40:18 ERROR ReceiverTracker: Deregistered receiver for stream 0: Restarting receiver with delay 2000ms: Socket data stream had no more data
16/03/11 22:40:18 INFO ReceiverSupervisorImpl: Stopped receiver 0
16/03/11 22:40:18 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
16/03/11 22:40:20 INFO JobScheduler: Added jobs for time 1457757620000 ms
16/03/11 22:40:20 INFO JniLoader: successfully loaded C:\Users\Vikas\AppData\Local\Temp\jniloader2012609788176054398netlib-native_ref-win-x86_64.dll
negative
16/03/11 22:40:20 INFO ReceiverSupervisorImpl: Starting receiver again
16/03/11 22:40:20 INFO ReceiverTracker: Registered receiver for stream 0 from 192.168.1.208:53206
16/03/11 22:40:20 INFO ReceiverSupervisorImpl: Starting receiver
16/03/11 22:40:20 INFO ReceiverSupervisorImpl: Called receiver onStart
16/03/11 22:40:20 INFO ReceiverSupervisorImpl: Receiver started again
16/03/11 22:40:20 INFO SocketReceiver: Connecting to 192.168.1.182:9999
16/03/11 22:40:21 INFO SocketReceiver: Connected to 192.168.1.182:9999
16/03/11 22:40:21 INFO SocketReceiver: Closed socket to 192.168.1.182:9999
16/03/11 22:40:21 WARN ReceiverSupervisorImpl: Restarting receiver with delay 2000 ms: Socket data stream had no more data
16/03/11 22:40:21 INFO ReceiverSupervisorImpl: Stopping receiver with message: Restarting receiver with delay 2000ms: Socket data stream had no more data:
16/03/11 22:40:21 INFO ReceiverSupervisorImpl: Called receiver onStop
16/03/11 22:40:21 INFO ReceiverSupervisorImpl: Deregistering receiver 0
16/03/11 22:40:21 ERROR ReceiverTracker: Deregistered receiver for stream 0: Restarting receiver with delay 2000ms: Socket data stream had no more data
4: Run 5: Debug 6: TODO Terminal Event Log
Compilation completed successfully in 29s 724ms (2 minutes ago)
761:1 CRLF: UTF-8:
I'm Cortana. Ask me anything.
10:41 PM
3/11/2016


T-Mobile
10:43
Android Controller
I'm waiting here: 9999
SiteLocalAddress: 192.168.1.182
#1 from /192.168.1.208:54513
replayed: Hey I am very happy and delighted
Hey I am very happy
delighted
SEND

6. Bibliography:

1) http://www.webdesignschoolsguide.com/library/10-things-we-couldnt-do-without-robots.html
2) http://www3.imperial.ac.uk/newsandeventspggrp/imperialcollege/newssummary/news_10-6-2014-11-42-22
3) http://www.burke.org/media/press/2013/04/burke-medical-research-institute-opens-new/57
4) http://www.mayoclinic.org/tests-procedures/robotic-surgery/care-at-mayo-clinic/treatment/prc-20013988