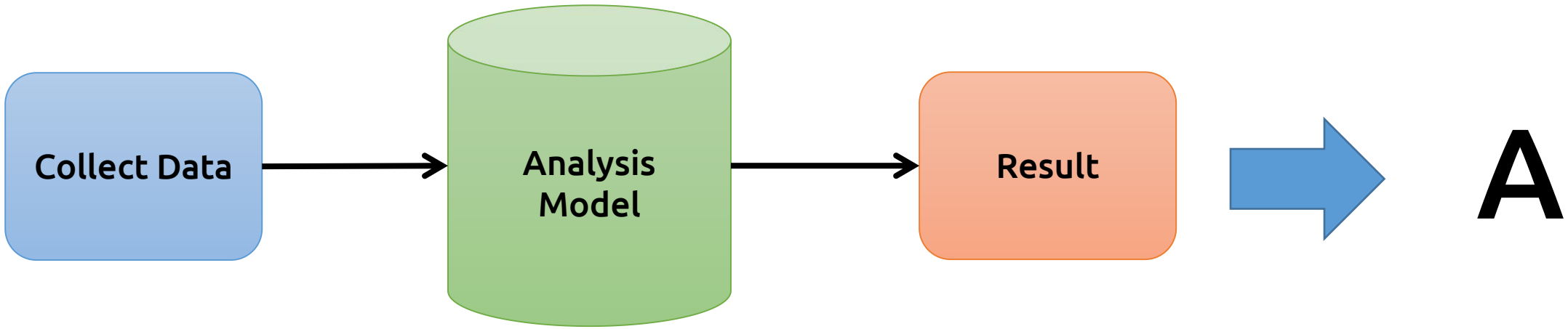
Data Collection

IS 698 Smart Home Health Analytics

H M Sajjad Hossain

Steps to getting A



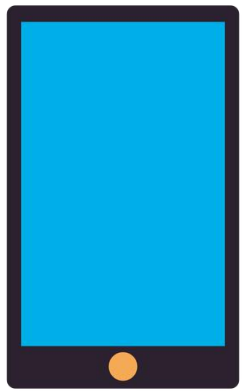
Outline

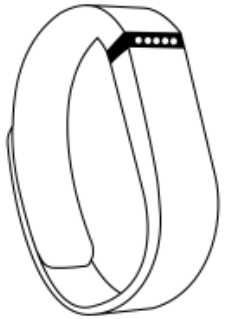
- Select problem domain
- What problem are you trying to solve?
- Sketch your plans
- What kind of data do you need?
- Which device should you choose?
- What kind of analysis you want to do?
- What is your final output?

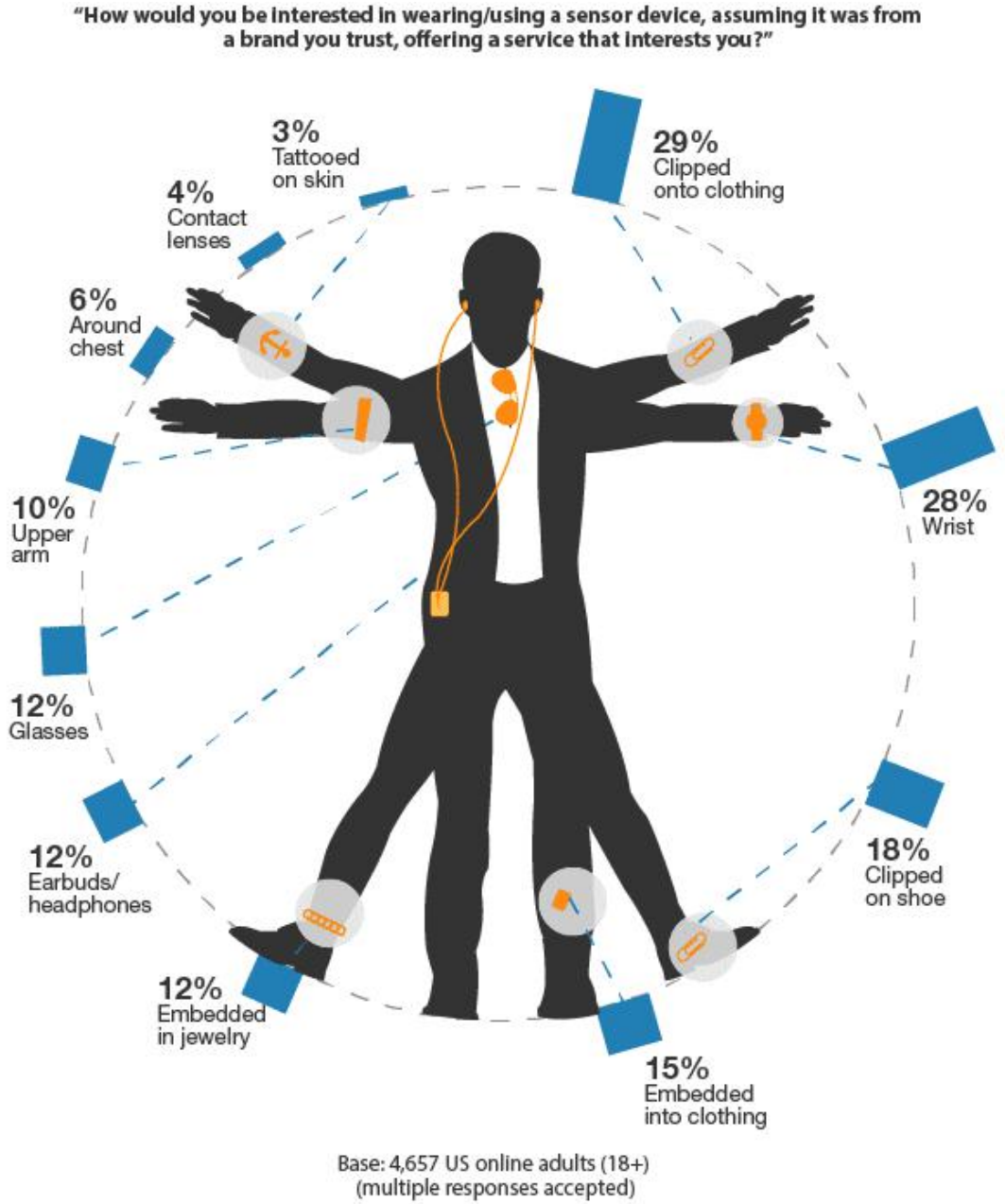
Sensors == Data

- Accelerometer, Gyroscope, Magnetometer
  - 3-axis (x,y,z)
- Galvanic Skin Response (GSR)
- Heart Rate sensor
- Step counter
- Motion sensor
- Raw Data: Raw sensor output
- Snapshot data: Step counts, Activity summary etc

Devices






"How would you be interested in wearing/using a sensor device, assuming it was from a brand you trust, offering a service that interests you?"
3% Tattooed on skin
29% Clipped onto clothing
4% Contact lenses
6% Around chest
10% Upper arm
28% Wrist
12% Glasses
12% Earbuds/ headphones
18% Clipped on shoe
12% Embedded in jewelry
15% Embedded into clothing
Base: 4,657 US online adults (18+)
(multiple responses accepted)
Source: North American Technographics® Consumer Technology Survey, 2013
97141
Source: Forrester Research, Inc.

Raw Data vs Snapshot Data

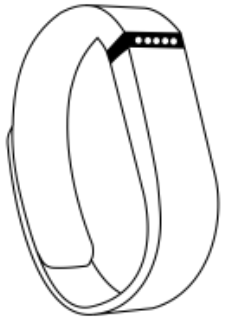
- Raw data is not processed
- Messy, Noisy and Huge
- Statistically interesting
- More control
- Multi dimensional analysis
- Appropriate feature extraction

- Snapshot data is processed raw data
- Cleaned, Noise free and Summerized
- Statistically interesting for supplementary analysis
- Less control
- Restricted analysis

Think Ahead

- Raw Data:
  - Smartphones
  - Smartwatches with OS
  - Research devices - Actigraph, Empatica etc.
- Most of the wearables.
  - Dashboard to see summary data

Which one provides Raw data?

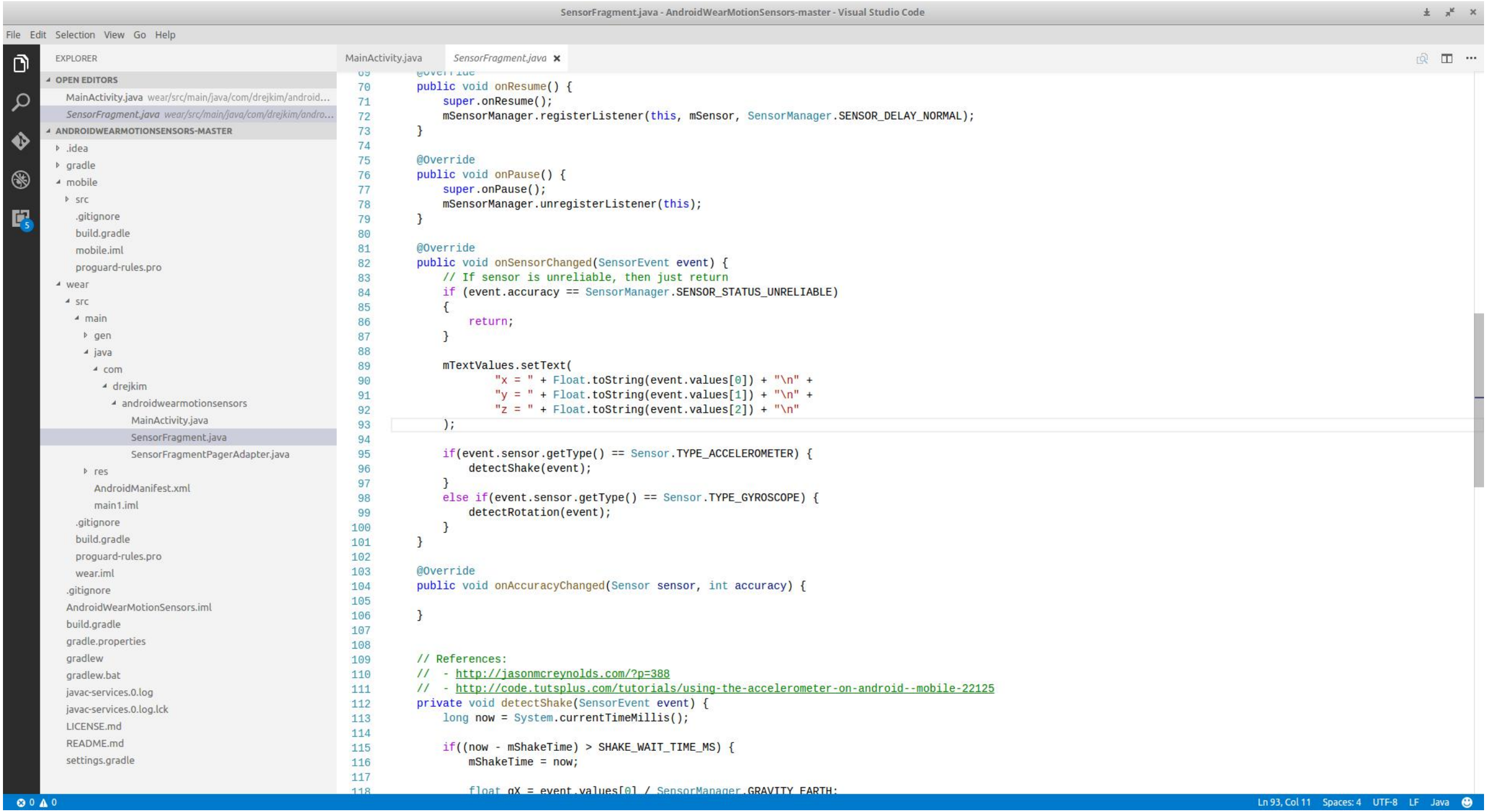


Available Apps

- Smartphones:
  - Android: Andro Sensor, Sensor Kinectics
  - iPhone: PowerSense
- Wearable:
  - Android: Sensor Data Logger (https://github.com/Steppschuh/Sensor-Data-Logger) -> No export functionality, Sensor Data Collector
  - iPhone: PowerSense

DIY

- Microsoft Band:
  - **Companion for Band**
  - Microsoft Band
- Android Watch:
  - Android Studio
  - Android SDK
  - https://github.com/pocmo/SensorDashboard
  - https://github.com/drejkim/AndroidWearMotionSensors


SensorFragment.java - AndroidWearMotionSensors-master - Visual Studio Code
File Edit Selection View Go Help
EXPLORER
OPEN EDITORS
MainActivity.java wear/src/main/java/com/drejkim/android...
SensorFragment.java wear/src/main/java/com/drejkim/andro...
ANDROIDWEARMOTIONSENSORS-MASTER
.idea
gradle
mobile
src
.gitignore
build.gradle
mobile.iml
proguard-rules.pro
wear
src
main
gen
java
com
drejkim
androidwearmotionsensors
MainActivity.java
SensorFragment.java
SensorFragmentPagerAdapter.java
res
AndroidManifest.xml
main1.iml
.gitignore
build.gradle
proguard-rules.pro
wear.iml
.gitignore
AndroidWearMotionSensors.iml
build.gradle
gradle.properties
gradlew
gradlew.bat
javac-services.0.log
javac-services.0.log.lck
LICENSE.md
README.md
settings.gradle
MainActivity.java
SensorFragment.java
@Override
public void onResume() {
super.onResume();
mSensorManager.registerListener(this, mSensor, SensorManager.SENSOR_DELAY_NORMAL);
}
@Override
public void onPause() {
super.onPause();
mSensorManager.unregisterListener(this);
}
@Override
public void onSensorChanged(SensorEvent event) {
// If sensor is unreliable, then just return
if (event.accuracy == SensorManager.SENSOR_STATUS_UNRELIABLE)
{
return;
}
mTextValues.setText(
"x = " + Float.toString(event.values[0]) + "\n" +
"y = " + Float.toString(event.values[1]) + "\n" +
"z = " + Float.toString(event.values[2]) + "\n"
);
if(event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
detectShake(event);
}
else if(event.sensor.getType() == Sensor.TYPE_GYROSCOPE) {
detectRotation(event);
}
}
@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
}
// References:
// - http://jasonmcreynolds.com/?p=388
// - http://code.tutsplus.com/tutorials/using-the-accelerometer-on-android--mobile-22125
private void detectShake(SensorEvent event) {
long now = System.currentTimeMillis();
if((now - mShakeTime) > SHAKE_WAIT_TIME_MS) {
mShakeTime = now;
float gX = event.values[0] / SensorManager.GRAVITY_EARTH;
0 0
Ln 93, Col 11 Spaces: 4 UTF-8 LF Java

Analysis

- Feature extraction.
  - Laptop?
  - University?
- Time domain features: Mean, Standard deviation, Variance etc.
- Frequency domain features: FFT, MFCC etc.
- Data annotation
- Tools: WEKA, R, Python

Do not just feed the data to your classification model. Understand and visualize the data, find important features. (https://blogs.msdn.microsoft.com/microsoftrservertigerteam/2017/03/23/feature-engineering-using-r/)

Thank You