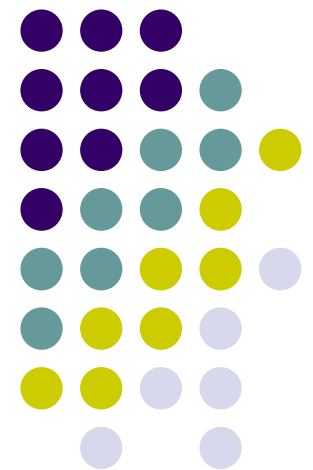


**CS 403X Mobile and Ubiquitous
Computing
Lecture 15: Making Apps
Intelligent/Machine Learning**

Emmanuel Agu



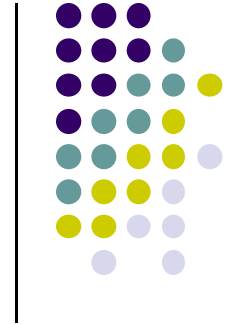


Making Apps Intelligent (Sensors Inference & Machine Learning)

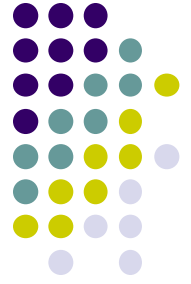


My Goals in this Section

- If you know machine learning
 - Set off light bulb
 - Projects involving ML?
- If you don't know machine learning
 - Get general idea, how it's used
- Knowledge will also make papers easier to read/understand

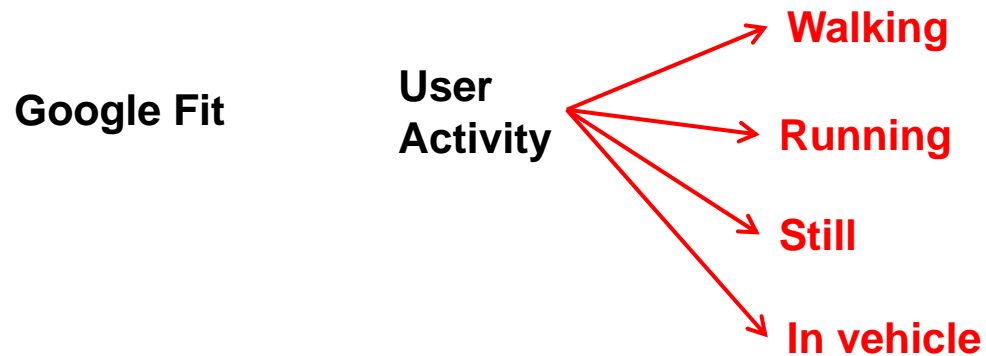
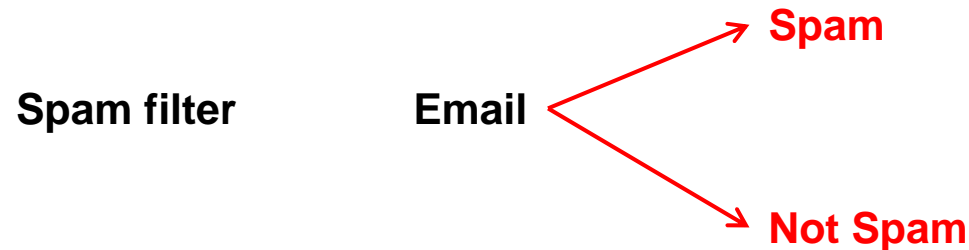


Intuitive Introduction to Classification/Supervised Machine Learning



Classification

- Classification is type of machine learning used a lot in Ubicomp
- Classification? determine which class a sample belongs to
- Examples:





Classifier

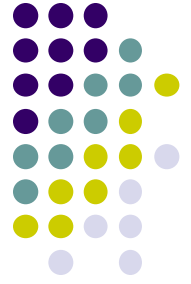
- Spam filter, Google Fit run a classifier
- Classifier:
 - Inspects new sample, decides which class
 - Created using example-based approach
- Classifier created using supervised machine learning
 - Supervised: labelled data as input
 - Examples of each class => generate rules to categorize new samples
 - **E.g:** Examples of spam email, non-spam email => generate rules to categorize new email





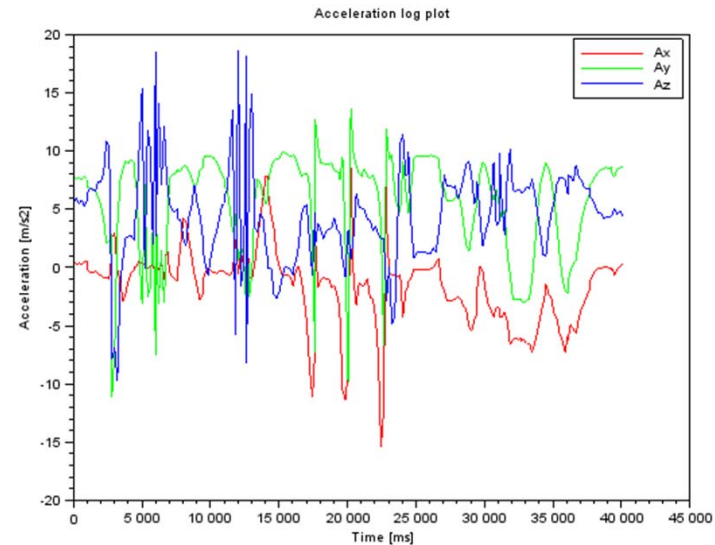
Explaining Classification/Supervised Learning using Activity Recognition

Activity Recognition



- Want app to detect when user is performing any of the following 6 activities

- Walking,
- Jogging,
- Ascending stairs,
- Descending stairs,
- Sitting,
- Standing

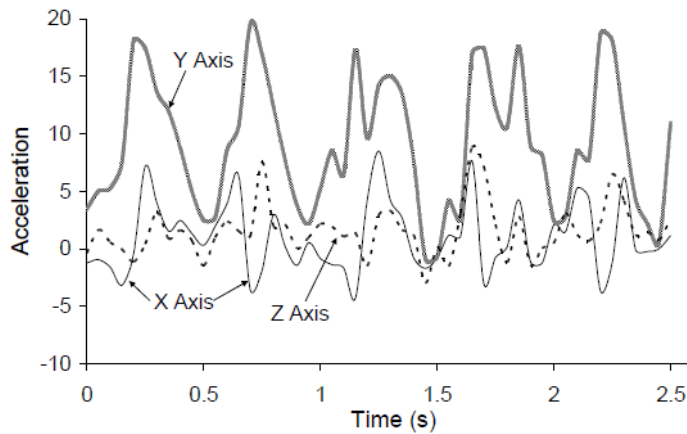


- **Approach:** Classifier to decide user activity based on accelerometer readings

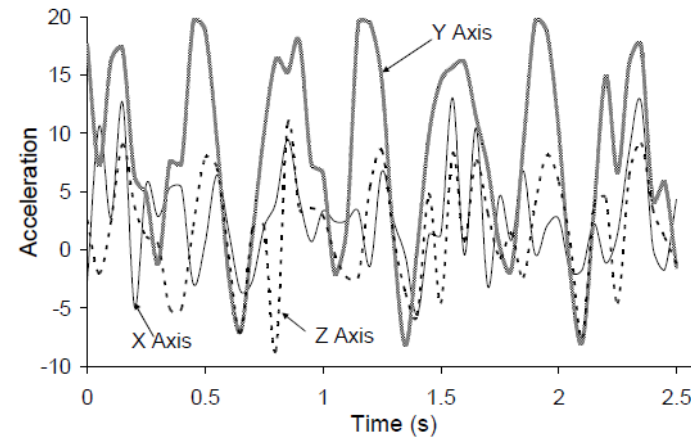
Example Accelerometer Data for Activities



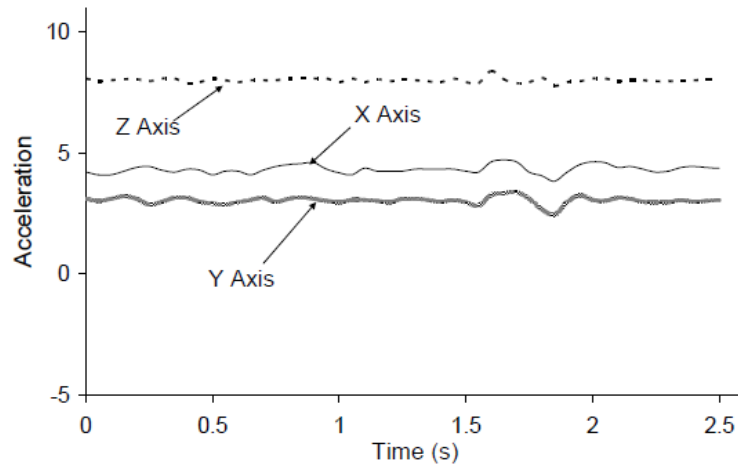
Step 1: Gather lots of example accelerometer data for each activity type



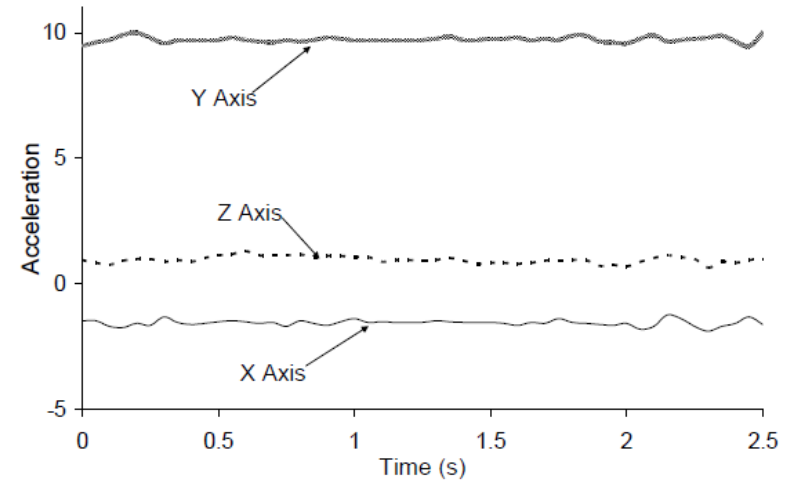
(a) Walking



(b) Jogging

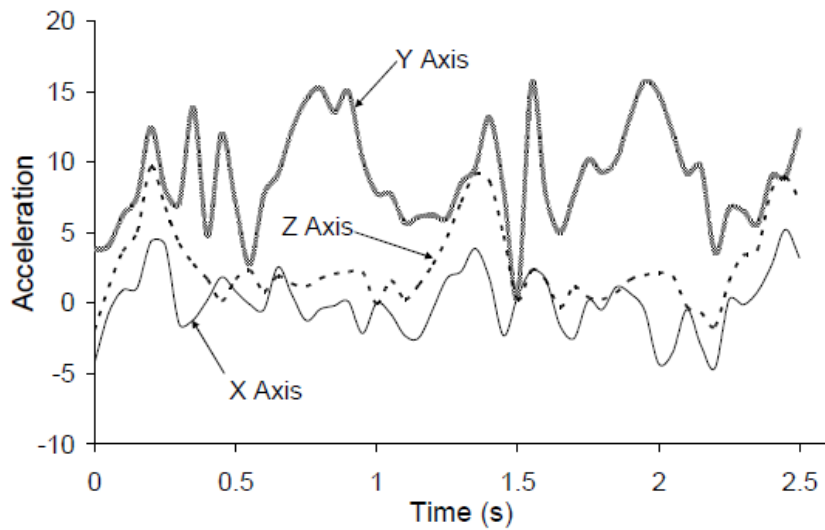


(e) Sitting

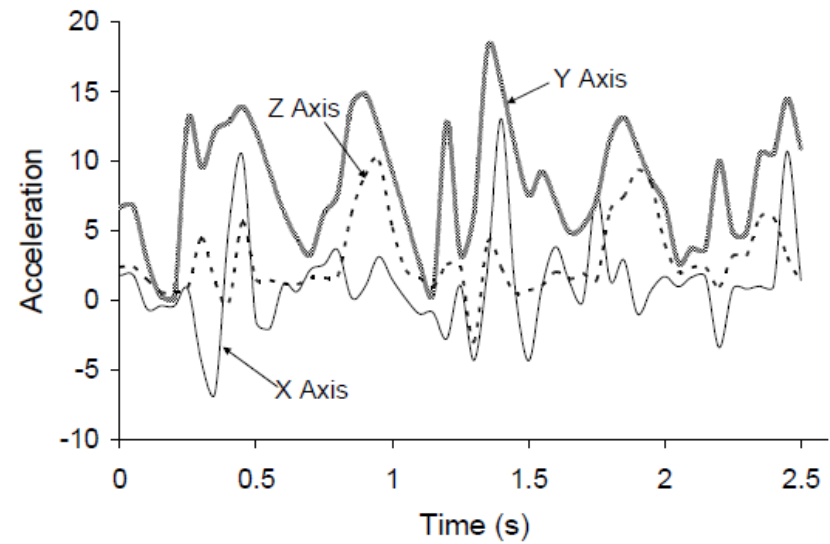


(f) Standing

Example Accelerometer Data for Activities



(c) Ascending Stairs

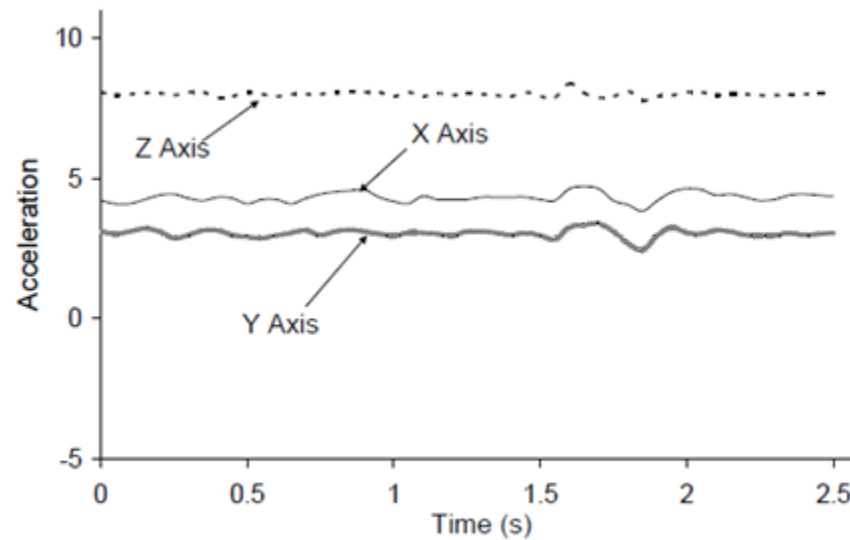


(d) Descending Stairs



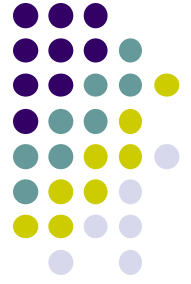
Gathering Accelerometer Data

- Can write simple app that retrieves accelerometer data while user is doing each of 6 activities (1 at a time)
- Label each data with activity performed.
 - E.g. label the following data as sitting



(e) Sitting

Funf (funf.org)



- Can also download, FUNF app from MIT to gather data
- Continuously collects user data in background:
 - Accelerometer readings
 - Phone calls
 - SMS messages, etc
- Simple to use:
 - Download app,
 - Check off sensors to log (e.g. accelerometer)



The screenshot shows the 'Funf In A Box: Android Application Set Up Form'. The form is titled 'funfinabox' and is divided into several sections:

- General (displayed in app):** Fields for App Name, Contact (Email), and Description.
- General (not displayed in app):** Fields for Your Name, Your Email, Organization Name, and Location.
- Configuration:** A note stating: "The following are the default data collection and configuration settings. They can be modified at: Dropbox\Funf In A Box\[Your App Name]\config\funf_config.json."
- Device:** Checkboxes for Android Info every [] seconds, Battery Info every [] seconds, Hardware Info every [] seconds, Mobile Network Info every [] seconds, and Time Offset every [] seconds.
- Device Interaction:** Checkboxes for Audio Files every [] seconds and Browser Bookmarks every [] seconds.

At the bottom, the text reads: 'Funf In A Box: Android Application Set Up Form'.



Step 2: Run Study to Gather Example Data

- Data collected from many (e.g. 30) subjects
- Users run Funf in their phones while performing each activity
 - Perform each of 6 activities (walking, sitting,.. Etc)
- Accelerometer data collected every 50ms
- Funf pushes data to dropbox, download data
- Now have 30 examples of each activity

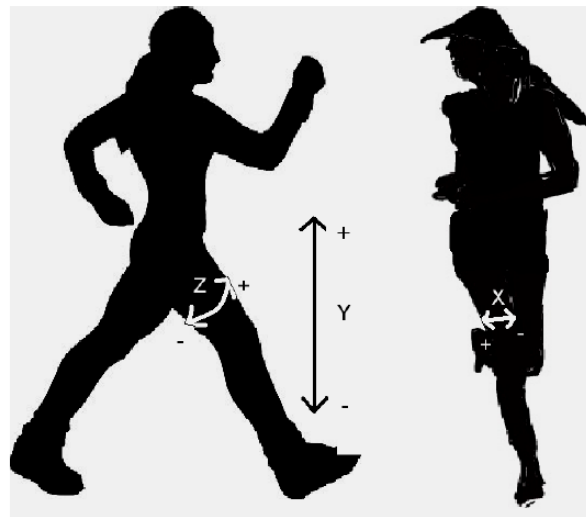
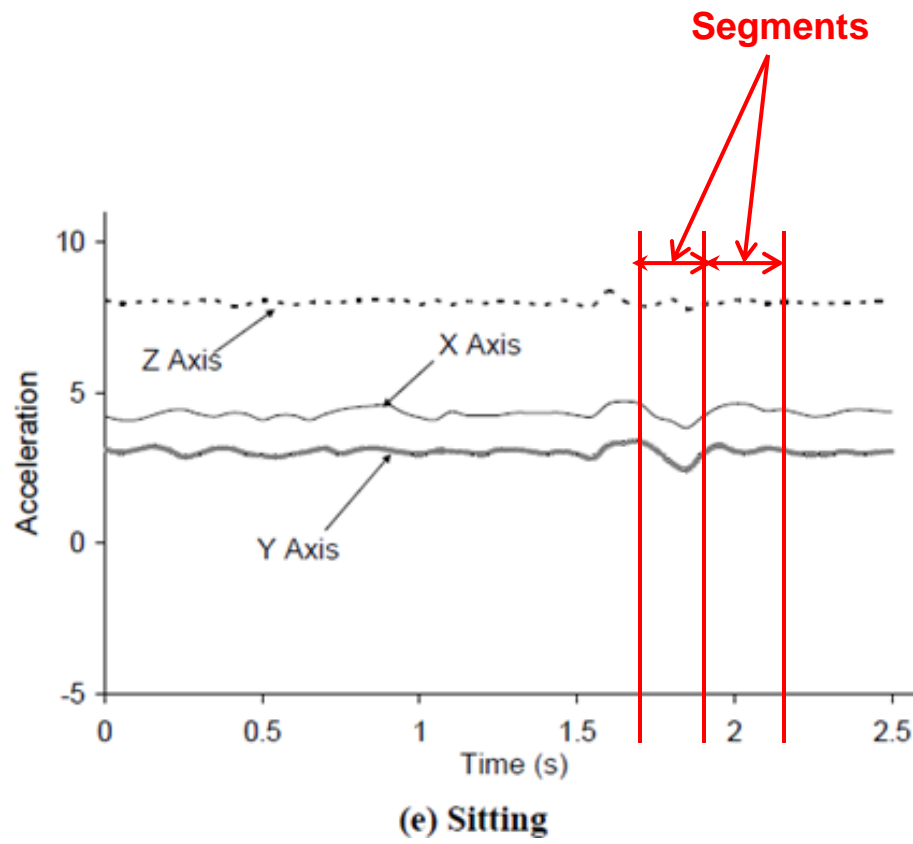


Figure 1: Axes of Motion Relative to User

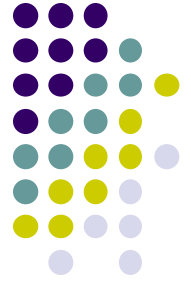


Segment Data (Windows)

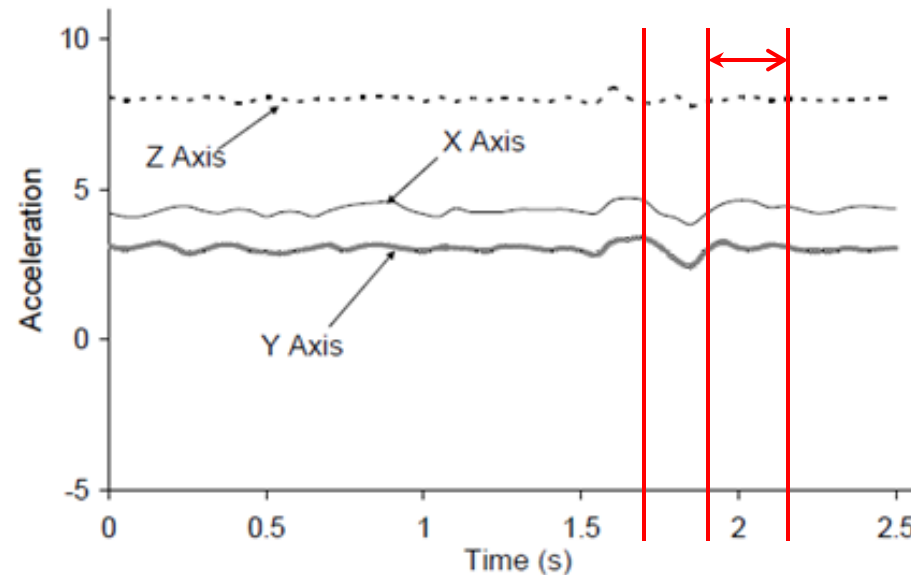
- Divide raw time-series data divided into segments (e.g. 10 seconds)



Compute Features



- Within segments, compute features
- **Features:** Functions computed on accelerometer data, captures important accelerometer characteristics
- **Examples:** min-max values within segment, magnitude within segment, standard deviation, moving average



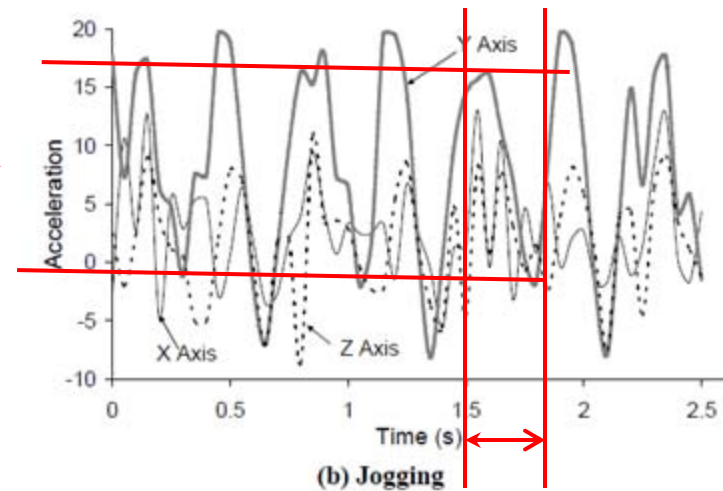
(e) Sitting



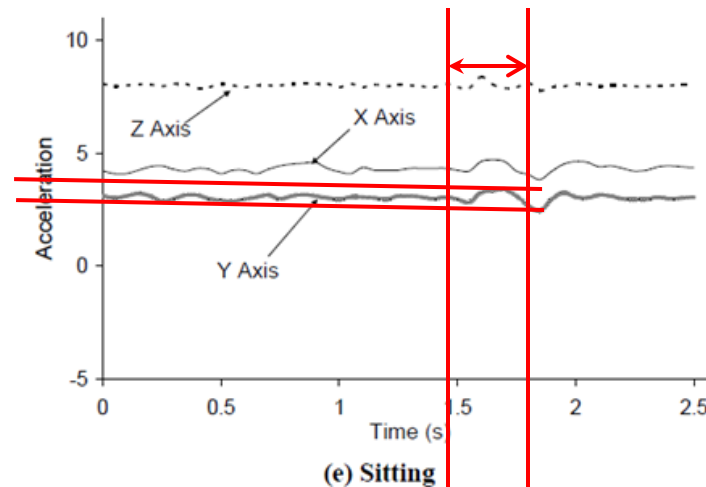
Compute Features

- **Important:** For given feature formula, each of activities should yield a different range of values
- **E.g:** Min-max Y axis range feature

Large min-max
for jogging



Small min-max
for jogging





Feature Computation

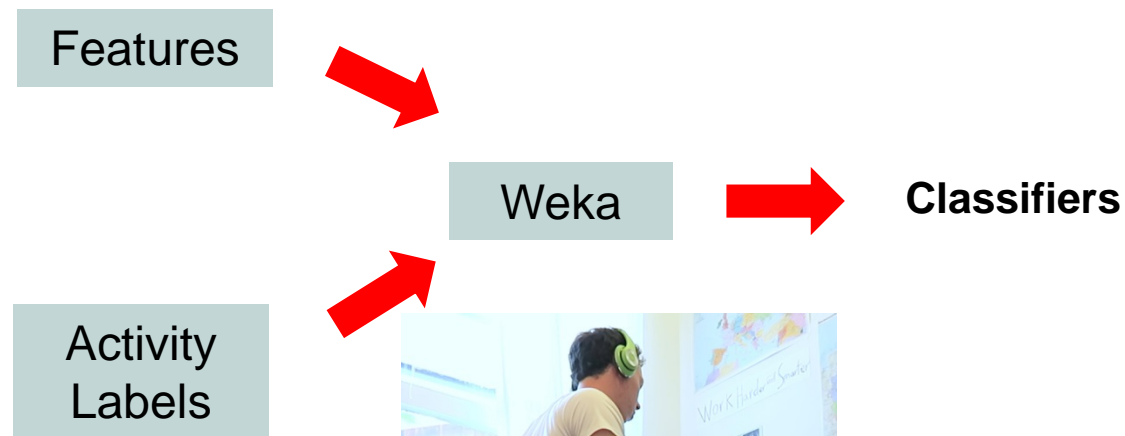
Calculate many different features

- Average[3]: Average acceleration (for each axis)
- Standard Deviation[3]: Standard deviation (for each axis)
- Average Absolute Difference[3]: Average absolute difference between the value of each of the 200 readings within the ED and the mean value over those 200 values (for each axis)
- Average Resultant Acceleration[1]: Average of the square roots of the sum of the values of each axis squared $\sqrt{(x_i^2 + y_i^2 + z_i^2)}$ over the ED
- Time Between Peaks[3]: Time in milliseconds between peaks in the sinusoidal waves associated with most activities (for each axis)
- Binned Distribution[30]: We determine the range of values for each axis (maximum – minimum), divide this range into 10 equal sized bins, and then record what fraction of the 200 values fell within each of the bins.



Machine Learning

- Pull calculated features + activity labels into Weka (or other Machine learning Framework)

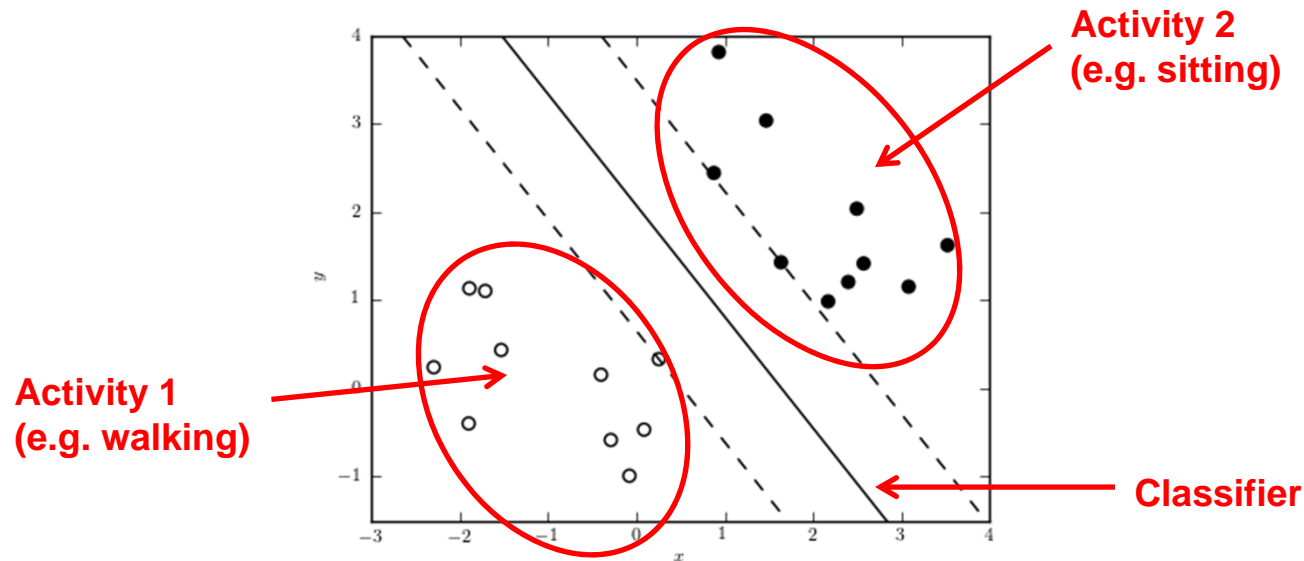


Done offline



What does Weka do?

- Features are just numbers
- Different values for different activities
- Weka figures out ranges corresponding to each activity
- Tries different classifier algorithms (SVM, Naïve Bayes, Random Forest, J48, etc)
- SVM example



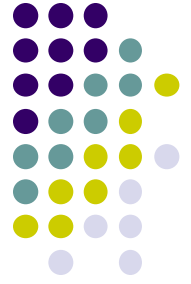


Accuracy of Classifiers

- Weka also reports accuracy of each classifier type

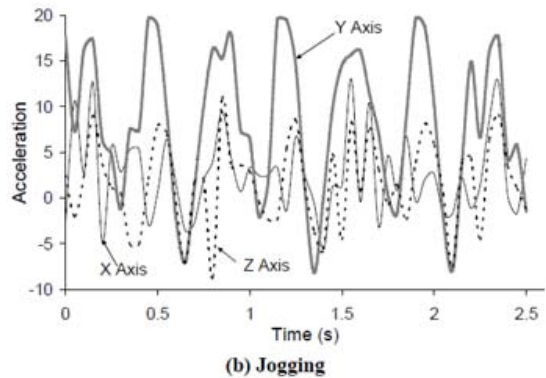
Table 2: Accuracies of Activity Recognition

	% of Records Correctly Predicted			
	J48	Logistic Regression	Multilayer Perceptron	Straw Man
Walking	89.9	<u>93.6</u>	91.7	37.2
Jogging	96.5	98.0	<u>98.3</u>	29.2
Upstairs	59.3	27.5	<u>61.5</u>	12.2
Downstairs	<u>55.5</u>	12.3	44.3	10.0
Sitting	<u>95.7</u>	92.2	95.0	6.4
Standing	<u>93.3</u>	87.0	91.9	5.0
Overall	85.1	78.1	<u>91.7</u>	37.2



Export Classifier from Weka

- Export classifiers as Java JAR file
- Run classifier in Android app
- Classifies new accelerometer patterns while user is performing activity => Guess (infer) what activity



**New accelerometer
Sample in real time**



**Classifier in
Android app**

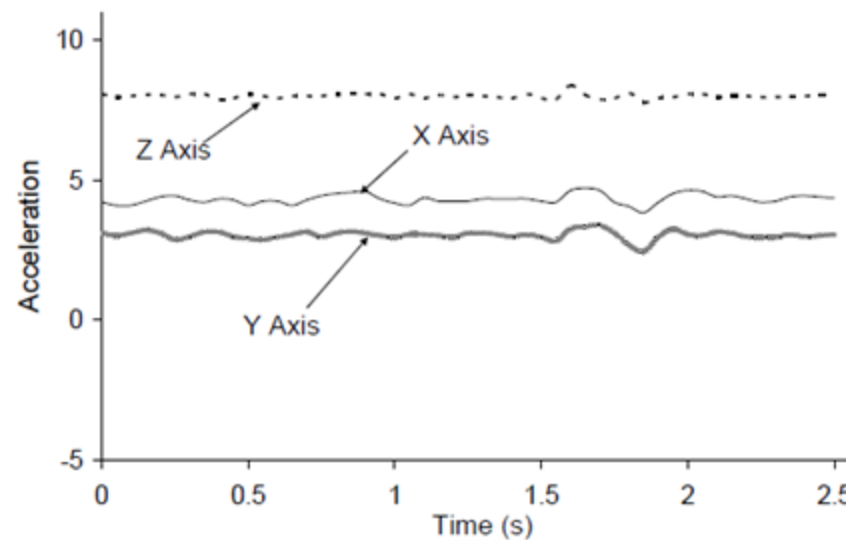


**Activity
(e.g. Jogging)**



What if you don't know Machine Learning

- Visually inspect accelerometer waveform, come up with rules by trial and error
- E.g. If (min-max range < threshold), activity = sitting



(e) Sitting

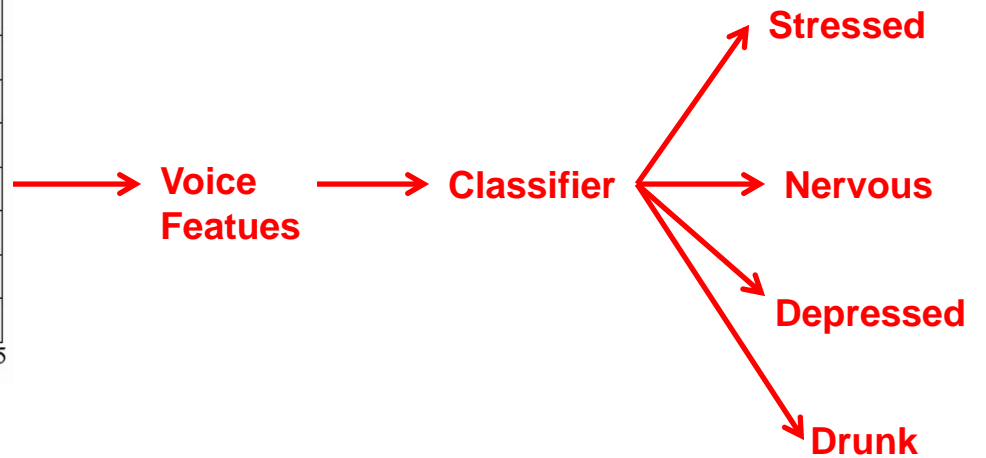
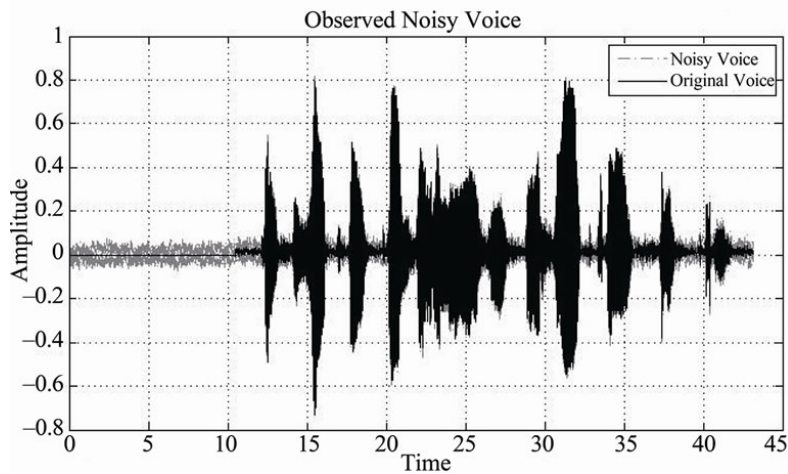


Concrete Examples of Classification



Voice Classification

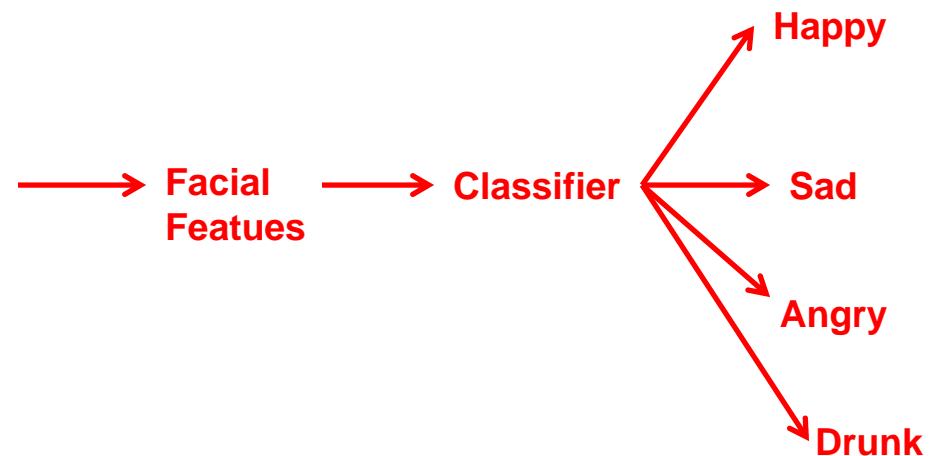
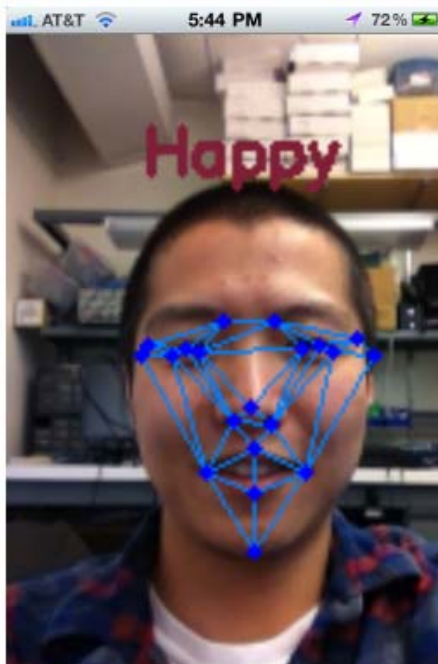
- Voice input from Phone microphone

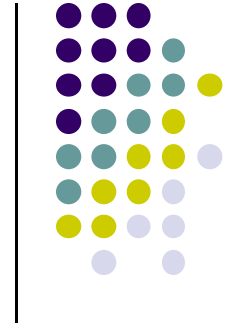




Facial Expression Classification

- Most of computer vision uses machine learning
- Classify camera images, to infer mood



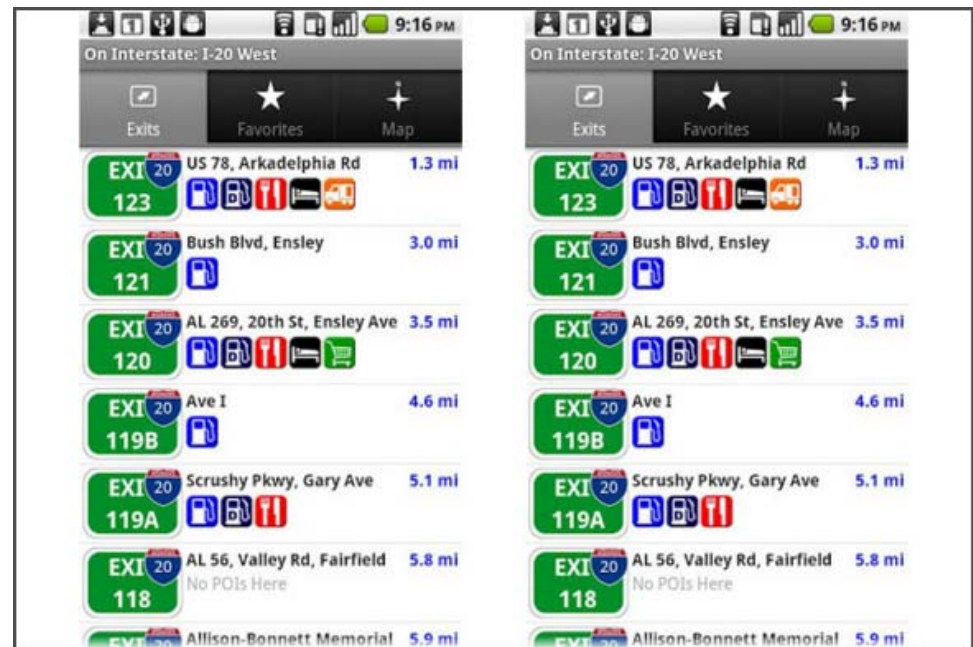


More Location-Aware Apps



iExit Interstate Exit Guide

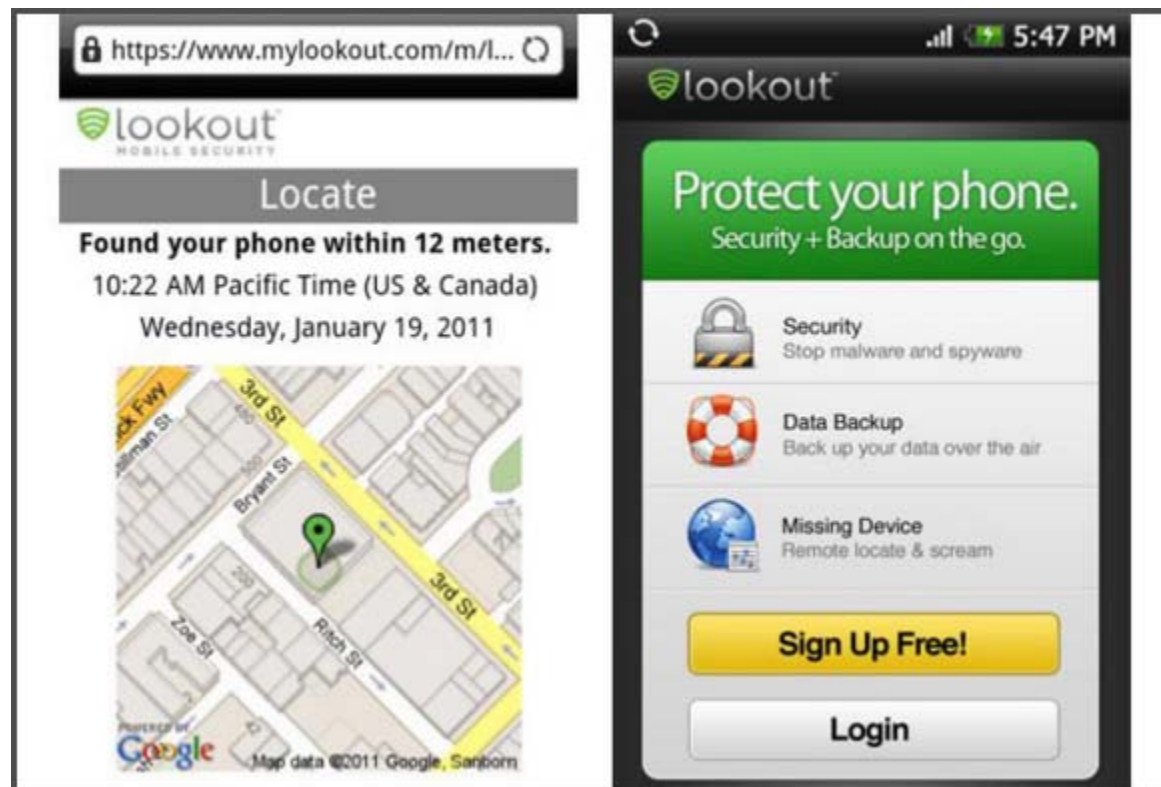
- Hungry while driving? Need to Pee?
- Tells you which restaurants, points of interest are available off each exit on the highway
- Not available in the US
- **What Android modules? For what?**
- **What stats to decide if this is tackling important problem?**

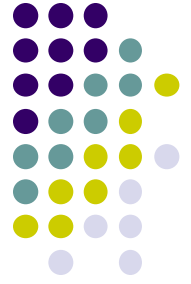




Lookout Security and AntiVirus

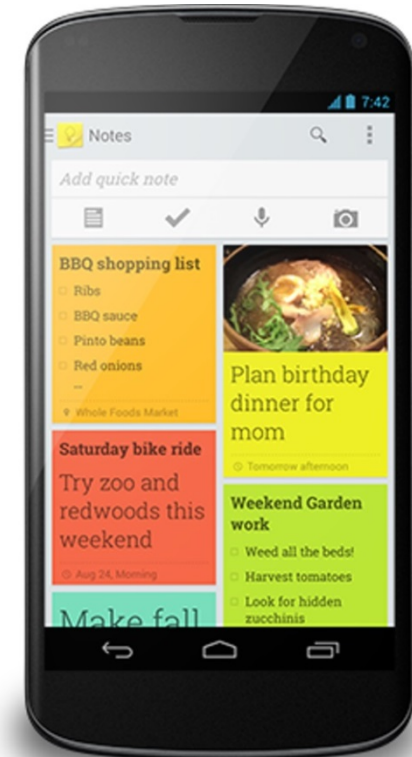
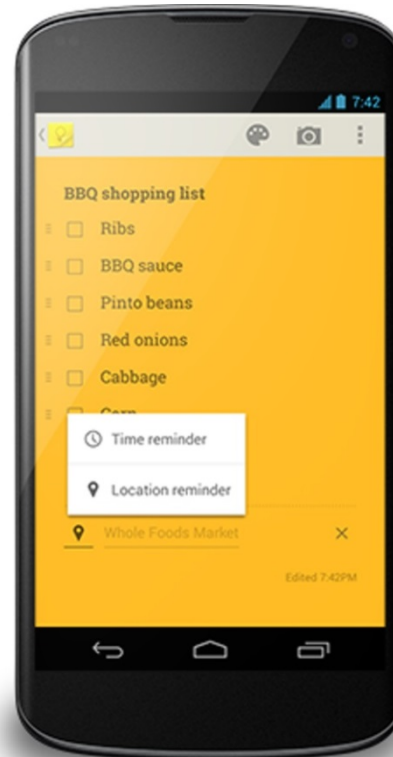
- Phone lost? Use Google GPS function to pinpoint phone location on map
- **What stats to decide if this is tackling important problem?**





Google Keep App

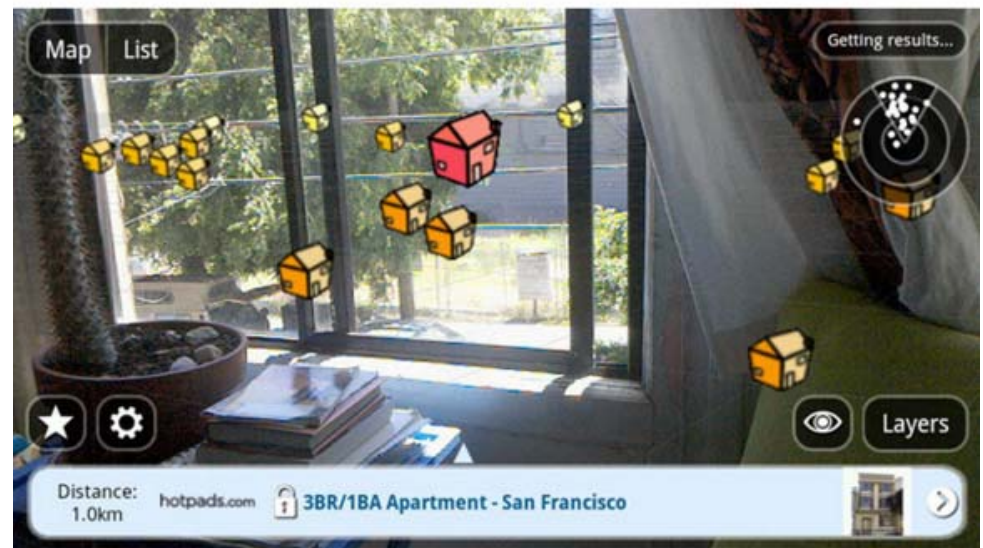
- Remind user of task at certain
 - Time
 - Location
- Powered by Google Now
- **How Android modules?**
For what?
- **What stats to decide if this is tackling important problem?**





Layar

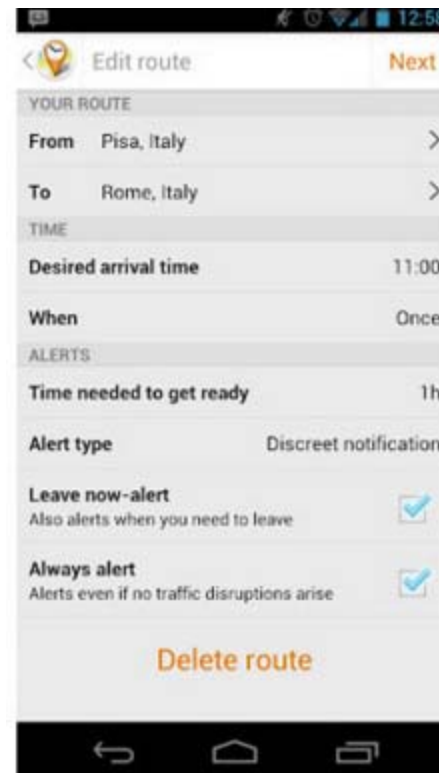
- Augmented Reality Browser
- Overlay information of location over real world
 - Information on restaurant you are at
 - Nearby apartments for rent
 - Tweets by people nearby
- **What layers would be useful for WPI students?**



NeverLate App



- Tells you when you have to leave Point A to get to Point B on time
- Factors in travel time, traffic, etc
- Sends notifications
- Not available in US
- **What Android modules? For what?**
- **What stats to decide if this is tackling important problem?**





Moves App

- Auto-track
 - Total time spent on various activities taken through the day
 - Timeline of activities, places visited, time spent
- Project idea? Implement subset of functionality?
- **How Android modules? For what?**
- **What vertical specific user types would find this app useful?**





References

- John Corpuz, 10 Best Location Aware Apps
- Liane Cassavoy, 21 Awesome GPS and Location-Aware Apps for Android,
- Head First Android
- Android Nerd Ranch, 2nd edition
- Busy Coder's guide to Android version 6.3
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014