R.U.M.
Recognizing Unsafe Motions

By Ben Chu, Joshua Chung, Zain Khan
Team Intros

Ben Chu
Lead Product Manager

Joshua Chung
Lead ML/Data Scientist

Zain Khan
Lead Developer, Data Engineer
Problem Statement

- Current State of Alcohol Consumption
- When drinking, nearly half of American drinkers typically consume 4 or more alcoholic beverages
- Research shows individuals do not realize they are impaired over 50% of the time while drinking
- Highly Expensive Modern Solutions

https://www.greenfacts.org/en/alcohol/l-2/01-number-people-affected.htm#0
Solution

We aim to create an efficient machine learning model to classify inebriation levels utilizing accelerometer data.
Global Impact

- **2 billion** active alcohol consumers
- **6.4 billion** smartphone users in the world
- **$1.36 billion** breathalyzer market size by 2026

[links]
https://www.greenfacts.org/en/alcohol/l-2/01-number-people-affected.htm#0
https://www.oberlo.com/statistics/how-many-people-have-smartphones
Target Audience

- Young professionals transitioning from college into their careers
- Smartphone users at bars, birthdays, weddings, or anyplace looking to stay aware while drinking
Product Demo

R.U.M.

Recognizing Unsafe Motions
Highlights of Technical Approach

- Real-time processing of data
- Low computational cost
- Offline availability
- Privacy
Model Input/Output

**Input**
iPhone Accelerometer Data

**Output**
Predicted Intoxication

Walk indicates little to no drunken movements
Walk indicates drunken movements
Data Pipeline

1. Accelerometer data is collected in real time
2. Data is passed into coreML model for prediction
3. The prediction is output instantaneously to the user in-app
Our Training Data

- UC Irvine Study
- 13 Participants tracked
- Accelerometer data sampled at 40hz
- Alcohol consumption tracked by TAC via SCRAM ankle monitors every 30 minutes
  - TAC: Transdermal Alcohol Concentration, similar units as BAC
Noise Reduction with Low-Pass Filter

- Common in signal processing
- Dampen Noise
- Preprocessed training data with Python
- Implemented in-app with Swift
Training Stabilization Using Virtual Realignment

- Accelerometer data’s mean and variance are affected by the orientation of the phone
- Based on the autocorrelation, the data can be virtually realigned
- Unable to port to CoreML, citing dependencies issues

Photo Credit: https://www.mathworks.com/help/supportpkg/android/ref/accelerometer.html
Defining our Neural Network Architecture

- Neural Network
  - No Feature Engineering Required
  - Fast Predictions
  - Not very interpretable

- Design inspired by previous work using a regression model for this type of problem

- Considered to be state-of-the-art time-series model architecture

Source: tensorflow.org
Improving Hidden State Dynamics with Layer Normalization

- Layer normalization normalizes activation by through each training point using mean and standard deviation
- Stabilizes training and hidden state dynamics
- Does not depend on batch size
- Built as a custom layer to be ported to CoreML

Source: https://medium.com/syncedreview/facebook-ai-proposes-group-normalization-alternative-to-batch-normalization-fb0699bffe7
Learning Short and Long-Term Signals with Bi-Directional LSTMs

- Bi-Directional LSTMs
  - Learn short and long-term signals
  - Look at data in both directions in time
- Elastic-Net Regularization
  - Kernel, Bias, Activation regularization

Source: https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/
Focus in a Global Context using Soft-Attention

- Soft Attention allows the model to focus on certain areas of the time window
- Signals that are far apart timewise can still influence the output of the model, which LSTMs fail to do

Implementing Leave One Subject Out CV

- Alternative to standard K-Fold CV
  - Each user used a validation fold
  - Train new model on each combination of n-1 folds

- Early Stopping (10 epochs)

- Measures performance on unseen users

- Average Accuracy: 63.6%
- True Positive Rate: 84.5%
- False Negative Rate: 15.4%
Modeling Takeaways

- Realignment of axes transforms the data to have consistent mean and standard deviation
- Noise filtering and Layer Normalization lead to training stabilization and convergence
- Use LOOCV testing that mirrors deployment environment, rather than standard methods
  - Attempt to validate model in deployment environment as well
Product Building Takeaways

- MVP features are heavily dependent on implementation constraints
- Know the constraints of your machine learning environment before beginning model development
- Begin app development early to better ideate on key features
- Leverage existing codebases to ease learning curve
Future Work

With more funding

1. Purchase an Apple Developer’s License to publish RUM on the app store
2. Run controlled trials to create more training and testing data
3. Build out app UI and develop background inebriation detection
4. Develop online learning techniques to tune model to user’s previous data over time
Summary

- **Mission:** Create an accessible way to assess your inebriation level
- **Problem Statement:** Billions around the globe have no affordable and accessible solution to inebriation detection
- **Product Differentiation:** Making inebriation prediction free and easy to use with an offline classification model
- **Impact:** Anyone with an Apple smartphone can now detect their inebriation level within 10 steps.
Closing Takeaways

- State of the art model, without requiring internet
- Zero-cost accessible inebriation measurement tool
- Accurate application for a billion dollar industry
- Billions of global drinkers can check their inebriation
Acknowledgements

- Github user @tylerhutcherson for app tutorial and skeleton code
  - https://github.com/skafos/ActivityClassifier
- USC and Ohio State researchers for modeling inspiration
- Estimation of Blood Alcohol Concentration From Smartphone Gait Data Using Neural Networks
- Using gait symmetry to virtually align a triaxial accelerometer during running and walking
Acknowledgements (Continued)

- https://medium.datadriveninvestor.com/volleyball-serve-detection-on-iphone-12-using-core-ml-40e3b29f73d4
Visit rum-app.com to view our website!
Thanks!

Any Questions?

FAQ

contact.rum.app@gmail.com
Group Contributions

- Zain - EDA, iOS app development, Lead Data Engineer, Model Support/ Debugging
- Joshua - Lead Data Scientist, Architecture design and testing, Porting model to CoreML
- Ben Chu - iOS wireframing, App testing, Website Design and Creation, Slide Theme and development