

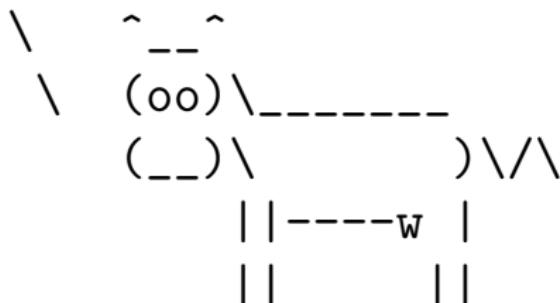
Music Recommendation

Jacob O'Bryant

20 April 2017

The Problem

/ It is inconvenient for people with \
| their own music collections to |
| |
| - organize their music with playlists |
\ - find new music /



The Problem



The Solution

Moodful next-track recommender



Implementation

The screenshot shows the Google Play Store interface. The top navigation bar includes the Google Play logo, a search bar with a magnifying glass icon, and categories like Home, Top Charts, and New Releases. On the left, a sidebar for 'My apps' is visible, with 'Shop' currently selected. Below it are links for Games, Family, and Editors' Choice. The main content area displays the details for the 'Vanilla Music' app by adrian.ulrich. The app's icon is a speaker with an orange ring, and its title is 'Vanilla Music'. It falls under the 'Music & Audio' category and is rated 'Everyone'. A note indicates compatibility with all devices. At the bottom right of the app's card is a green 'Installed' button. The footer of the page contains standard navigation icons.

Google Play

Search

Apps

Categories

Home

Top Charts

New Releases

My apps

Shop

Games

Family

Editors' Choice

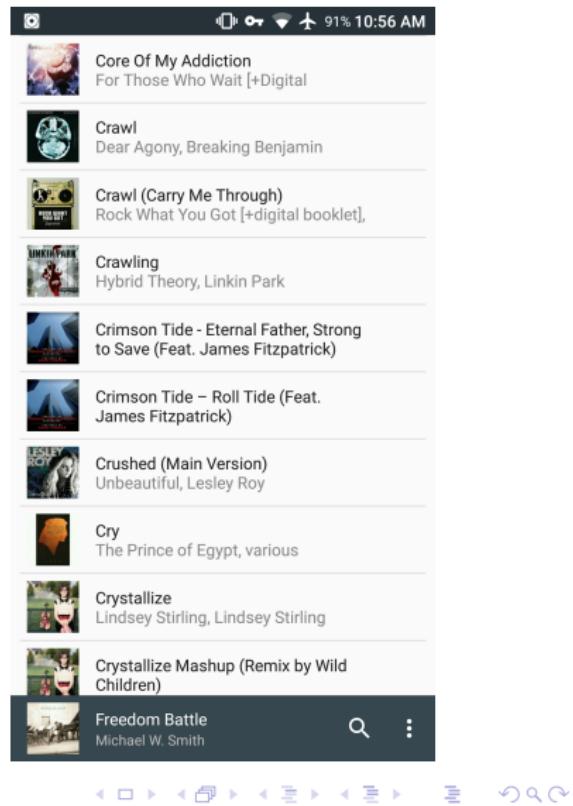
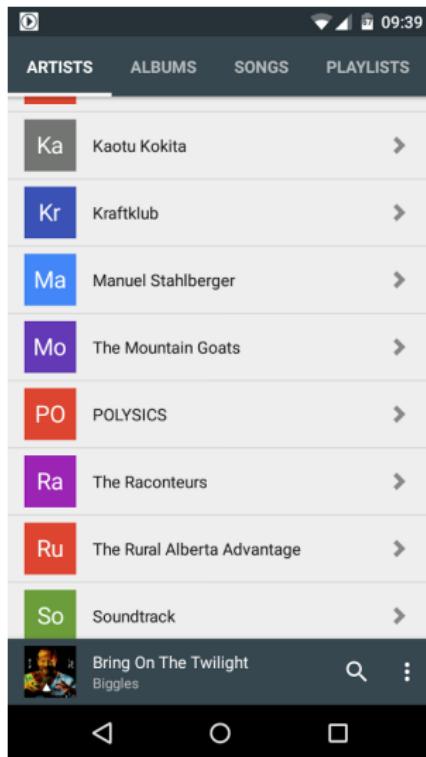
adrian.ulrich Music & Audio

Everyone

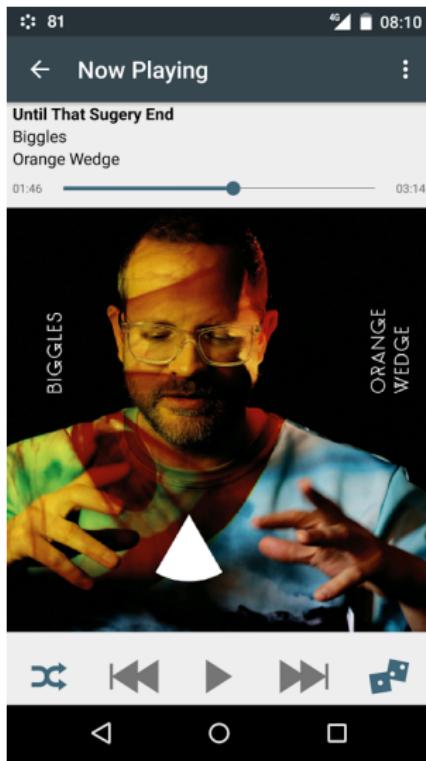
This app is compatible with all of your devices.

Installed

Implementation



Implementation



Implementation



Implementation (Moodless)

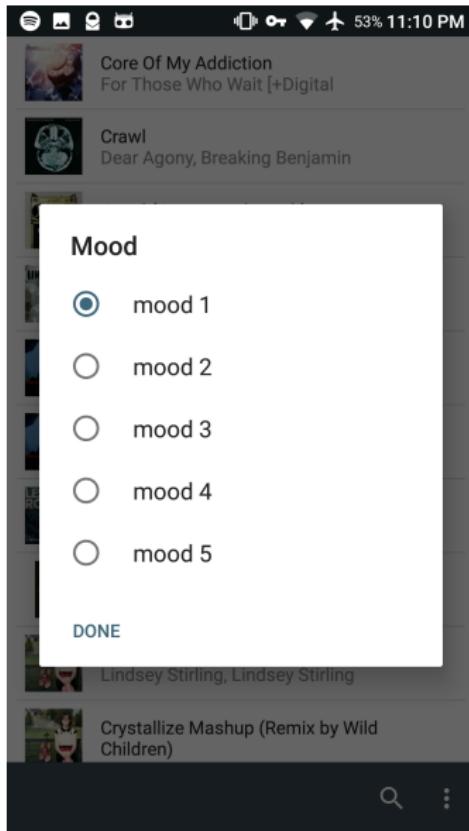
$$P = R_s C_s + (1 - C_s)(R_{al} C_{al} + (1 - C_{al})(R_{ar} C_{ar}))$$

Implementation (Moodless)

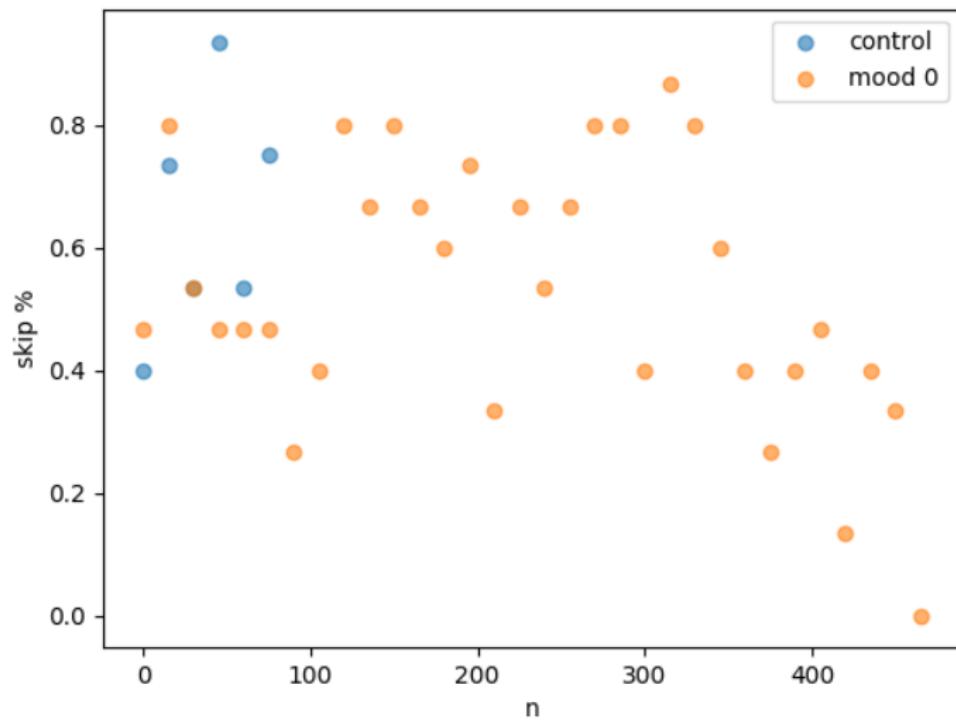
$$P = R_s C_s + (1 - C_s)(R_{al} C_{al} + (1 - C_{al})(R_{ar} C_{ar}))$$

$$C(n) = n/(n + a)$$

Implementation (Moodless)



Implementation (Moodless)



Implementation

A survey of music recommendation and possible improvements

Jacob O'Bryant

07 April 2017

Abstract

With the prevalence of the internet, mobile devices and commercial streaming music services, the amount of digital music available is greater than ever. Sorting through all this music is an extremely time-consuming task. Music recommendation systems search through this music automatically and suggest new songs to users. Music recommendation systems have been developed in commercial and academic settings, but more research is needed. The perfect system would handle all the user's listening needs while requiring

Implementation (Moodful)

Session-based collaborative filtering

Implementation (Moodful)

A Programmer's Guide to Data Mining

The Ancient Art of the Numerati



Chapters

- [1: Introduction](#)
- [2: Recommendation systems](#)
- [3: Item-based filtering](#)
- [4: Classification](#)
- [5: More on classification](#)
- [6: Naïve Bayes](#)
- [7: Unstructured text](#)
- [8: Clustering](#)

Implementation (Moodful)

Java :(

Lisp :)

Implementation (Moodful)

```
(ns reco.reco
  (:gen-class
    :implements [clojure.lang.IDeref]
    :state state
    :methods [[add_event [String, String,
                          String, boolean] void]
              [pick_next [] java.util.Map]]
    :init init
    :constructors {[java.util.Collection] []})))
```

Implementation (Moodful)

```
(defn -pick-next [this]
  (let [sessions (:sessions @(.state this))
        neighbors (filter #(and (not-empty (:recs %1))
                                 (> (:score %1) 0.4))
                           (map #(create-neighbor (first sessions)
                                     (rest sessions)))
        nearest (take k (sort-by :score neighbors))
        candidates (aggregate nearest)]
    (if (empty? candidates)
        (rand-nth (:songs @(.state this))))
        (:mdata (nth candidates (wrand (vec (map :score candidates
```

Implementation (Moodful)

```
; session 0
; same as rec.add_event("a", "a", "a", true, 2000)
(.add_event rec "a" "a" "a" true 2000)
(.add_event rec "b" "b" "b" false 2000)
(.add_event rec "c" "c" "c" false 2000)
(.add_event rec "d" "d" "d" false 2000)
(.add_event rec "e" "e" "e" false 2000)
(.add_event rec "f" "f" "f" false 2000)

; session 1
(.add_event rec "a" "a" "a" false 4000)
(.add_event rec "b" "b" "b" false 4000)
(.add_event rec "c" "c" "c" false 4000)
(.add_event rec "d" "d" "d" false 4000)
(.add_event rec "e" "e" "e" false 4000)
(.add_event rec "g" "g" "g" false 4000)
```

Implementation (Moodful)

```
; session 2
(.add_event rec "a" "a" "a" false 6000)
(.add_event rec "b" "b" "b" false 6000)
(.add_event rec "c" "c" "c" false 6000)
(.add_event rec "d" "d" "d" false 6000)
```

Implementation (Moodful)

Did it work?

Implementation (Moodful)

Did it work?

Yes! Sort of...

TODO

Exploration vs.
Exploitation



Freshness



Model-based CF



Shameless Self-Promotion

jacobobryant.com/ideas/music