Decisions as a Service (DaaS)

Audris Mockus & Pat Tendick
The University of Tennessee and Avaya
Overview

- Growing body of *interactive applications*
- 1.5M apps in the Apple App Store, more for Android
- Scale of usage can be massive, e.g., Facebook has 1.4B active users
- Opportunities to optimize using big data and *predictive analytics*
- Serious challenges make this difficult
- We propose a new approach that uses software engineering
- The approach is application centric and takes into account the different roles involved
Concepts

- **Interactive Applications**
  - Web, mobile apps, IVR, chat/messaging, gaming, infotainment
  - Unlike traditional programs (event driven, e.g.)

- **Predictive Analytics**
  - Predict outcomes given what we know
  - Use statistics, data mining, or machine learning

- **Next Best Action/Next Best Offer**
  - Identify the best action to take given what we know now
  - Next best offer – Select the best product or service to offer based on what we know

- **Analytical Tools** – Used to develop prediction methods using statistics, DM, or ML, e.g., “How do we predict whether a caller will hang up?”

- **Scoring Engines** – Render predictions or scores based on new data, e.g., “How likely is this caller to hang up?”

- **Decisioning Engines** – Render decisions based on complex criteria, e.g., “Should this person be offered an auto loan, and at what interest rate?”
Challenges

- **Distinct roles:**
  - *Application developer* – Understands what the application should do and how to implement it. Typically won’t know much about analytics or prediction.
  - *Data scientist* – Understands how to analyze data and develop predictors. Typically won’t know anything about how the application actually works.
  - *Business analyst* – Understands what applications are supposed to do from a business perspective. Doesn’t typically understand how apps work or how predictive models work.

- **Many possible goals and outcomes of interest:**
  - Customer satisfaction
  - Cost
  - Revenue

- An organization could have many applications, each making many kinds of decisions
- Goals could change, rendering previous predictions and application code obsolete
Objective

- Create a toolset, framework, and methodology
  - Separate concerns of analyst and developer
  - Effective, adaptive instrumentation and serialization
  - Applicable to parts of complex systems
  - Flexible schema descriptions for data extraction and marshalling

Considerations...
- Roles
- Cloud vs. premise based solutions
- API
- Integration across channels, platforms, applications, etc.
Solution

- Take an application-centric approach.
- Provide a simple API.
- Provide *recommendations* instead of predictions.
- E.g., rather than saying the probability that the caller will hang up is 0.8, recommend that the call be routed to an operator.
- The *application* requests recommendations about the decisions it has to make. This is where the process starts.
- Collect data synchronously at the time the application requests a prediction.
- Collect data automatically, so the developer doesn’t have to think about it.
- Automatically save the data to a big data repository for later analysis.
- Generate datasets for analysis from this data.
Proposed Solution

- Applications
  - Shopping
  - Add Review
  - Orders
  - Checkout

- Decision API
  - Decision Requests/Data
  - Recommendations

- Decision Broker
  - Datasets
  - Scoring Requests & Responses

- Data Store

- Scoring Engine
  - Package 1
  - Package 2
  - Package 3

- Statistical Tools/Data Mining/ML
  - R
  - SAS
  - MLlib

© 2015 Avaya Inc. All rights reserved.
Questions for the audience: best ways to evaluate?

- Dimensions for experimental design?
- What to evaluate?
- How to evaluate?
- Evaluate each roles or the separation of concerns among them?
  - Developer
  - Data scientist
  - Business analyst
- Where would this approach be most appropriate?
  - Channel, e.g. Web, mobile app, voice apps
  - Platforms, e.g. Java servlets, PHP, node.js
  - Applications, e.g. e-commerce, content provider, banking