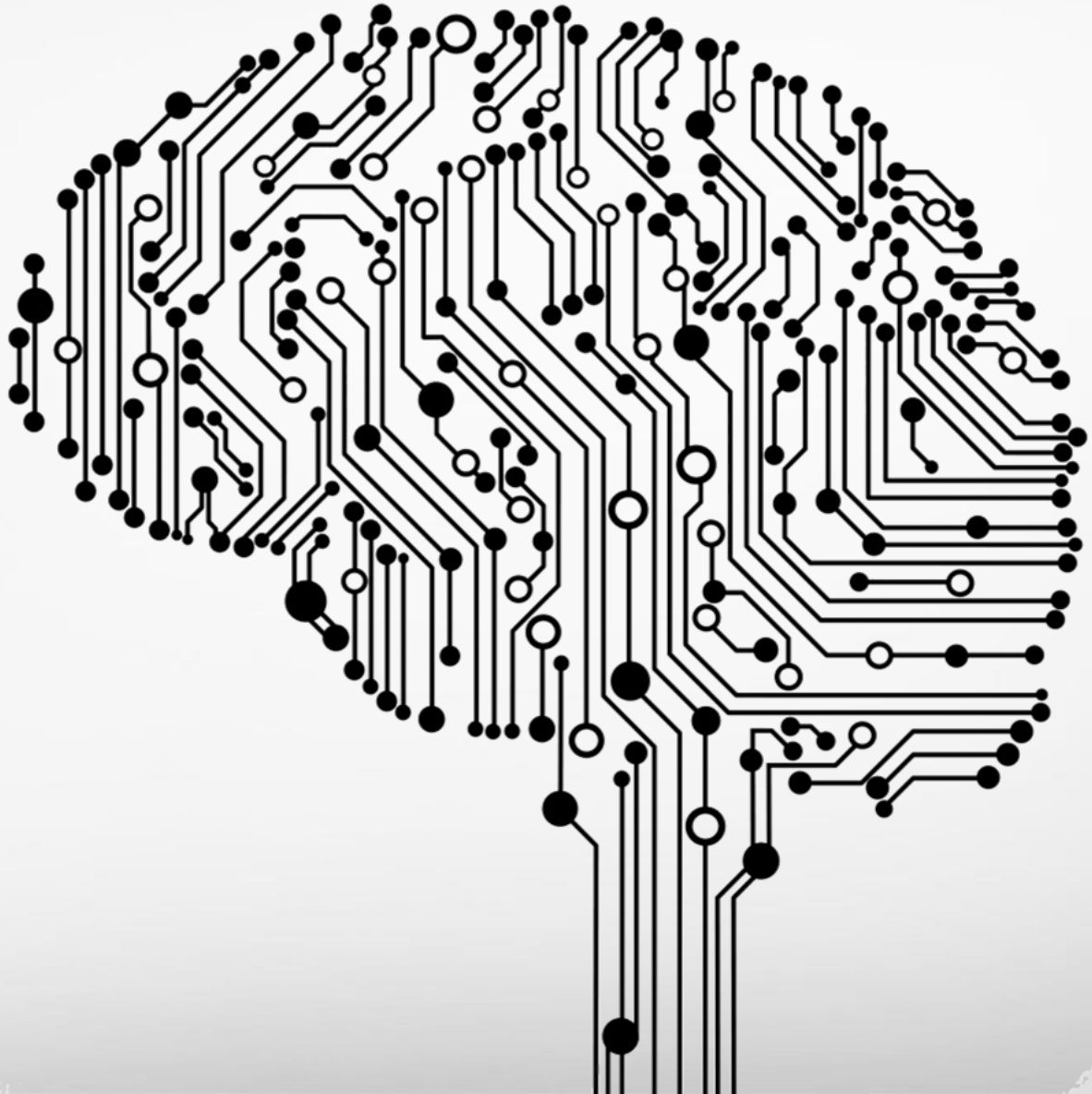
Microsoft

Custom Decision Service

A cloud-based, contextual decision-making API that sharpens with experience



The Decision Service in a nutshell

Repeatedly:

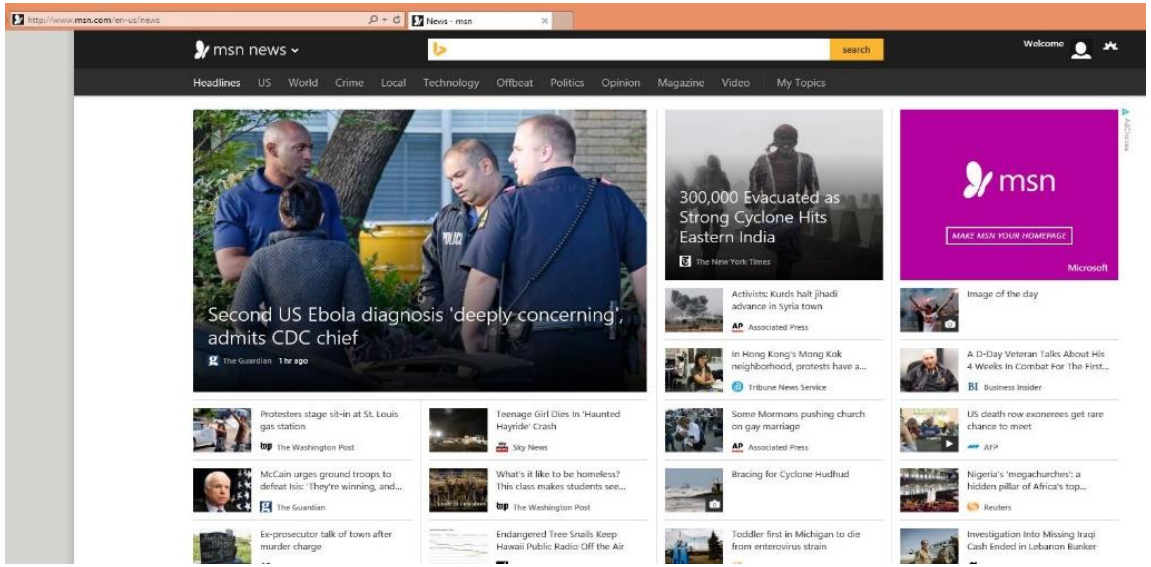
1. Sense the world.
2. Act.
3. Observe outcome.

Goal: Optimize (2) for (3)

Content Personalization

Repeatedly:

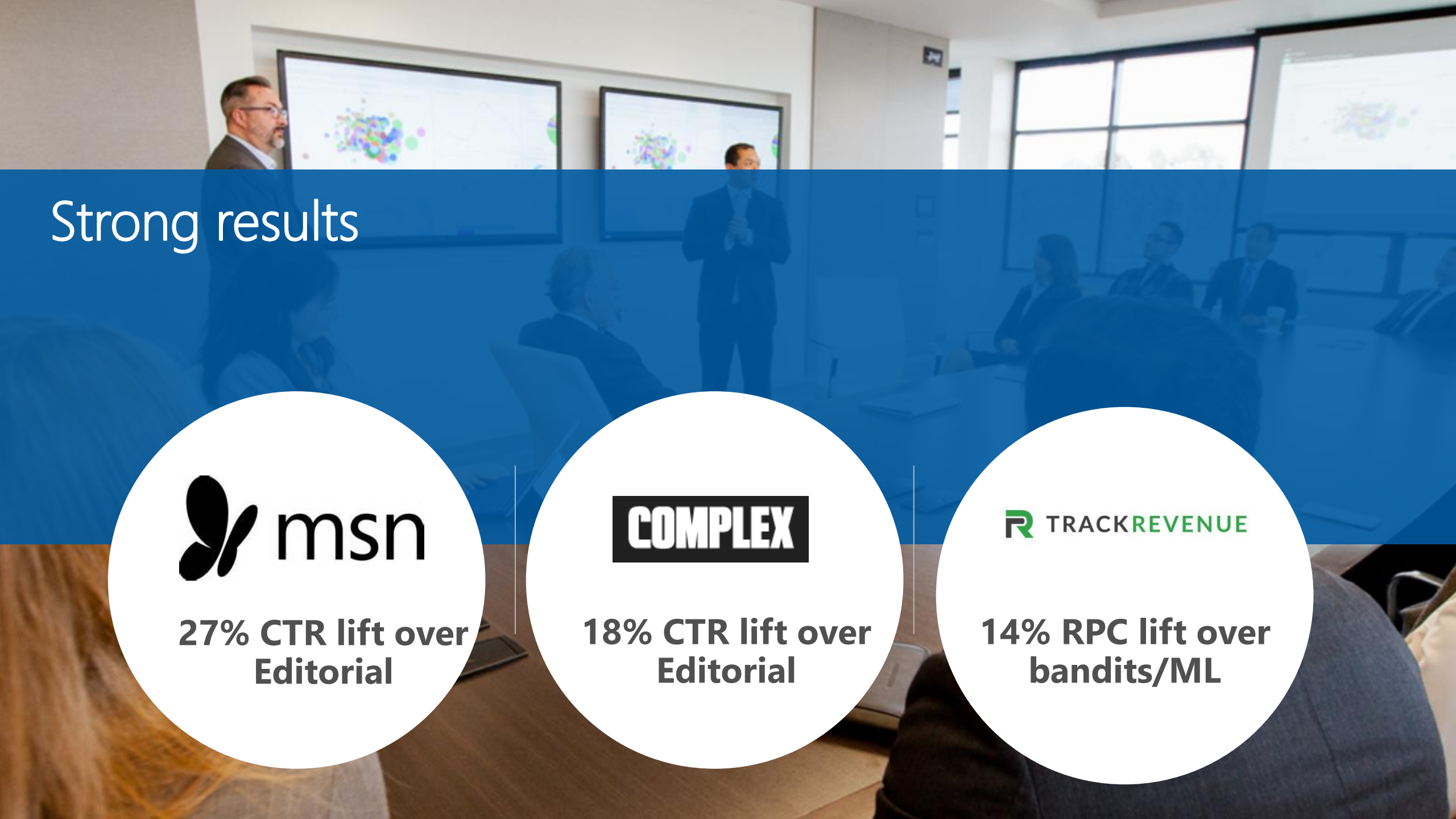
1. User **arrives** at complex.com
   → Sense
2. Decision Service **chooses** article ranking
   → Act
3. User **clicks** to content
   → Outcome



Decision Service

- Collects the right data needed to learn
- Explores alternatives
  - Avoid “User likes tech, only gets tech”
- Must be deployed to live system!

Strong results



msn

27% CTR lift over Editorial

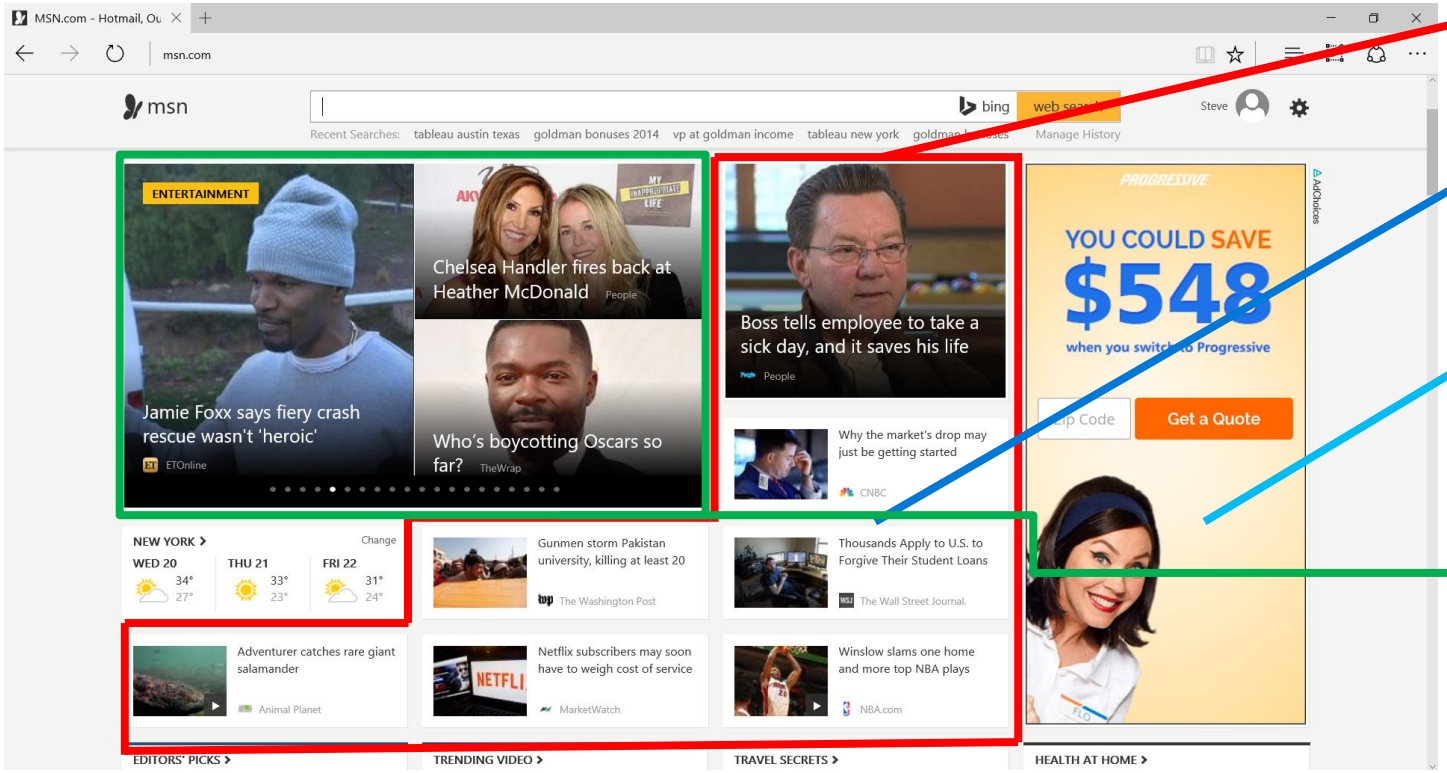
COMPLEX

18% CTR lift over Editorial

TRACKREVENUE

14% RPC lift over bandits/ML

MSN



Custom Decision Service

River has seen a 26.8% lift

Advertising click through lifted by 9%

Custom Decision Service

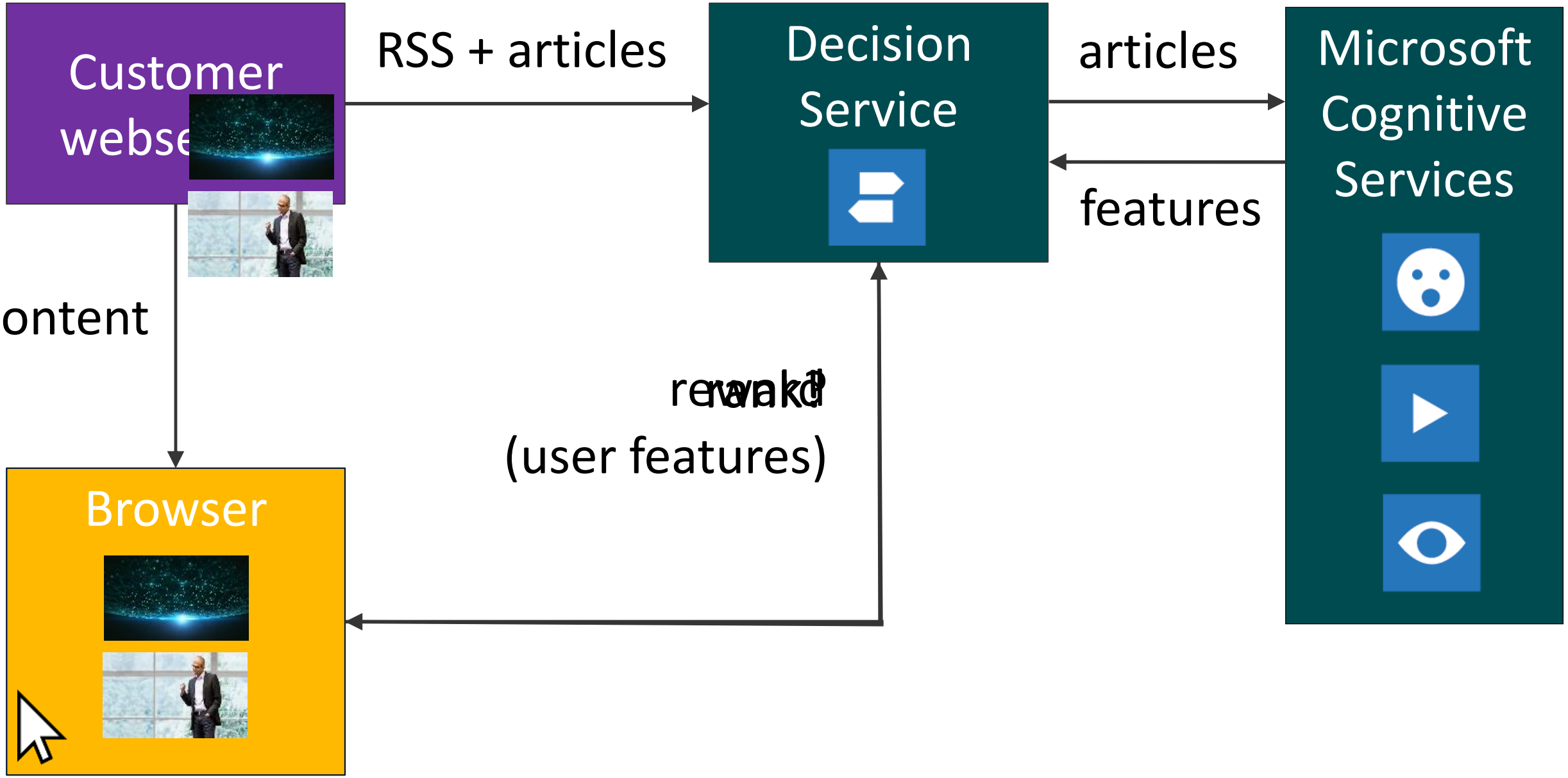
Content Personalization

Signup for free on https://ds.microsoft.com

1. Choose an application name

2. Register one or more RSS feeds

3. Provide Azure Storage account
   → Access logged data


Customer website
RSS + articles
Decision Service
articles
Microsoft Cognitive Services
features
content
reward
(user features)
Browser

Auto-featurization

HTML Extraction (SEO)

- title, description, text
- image

Cognitive Services

- Text: Entities, Sentiment
- Image: Vision, Emotion

User Features

- Geo location (reverse IP)
- Browser type
- Referrer

Customize Auto-featurization

Open Source

- https://github.com/Microsoft/mwt-ds/tree/master/Crawl
- Azure Functions + Azure Logic Apps using ARM template

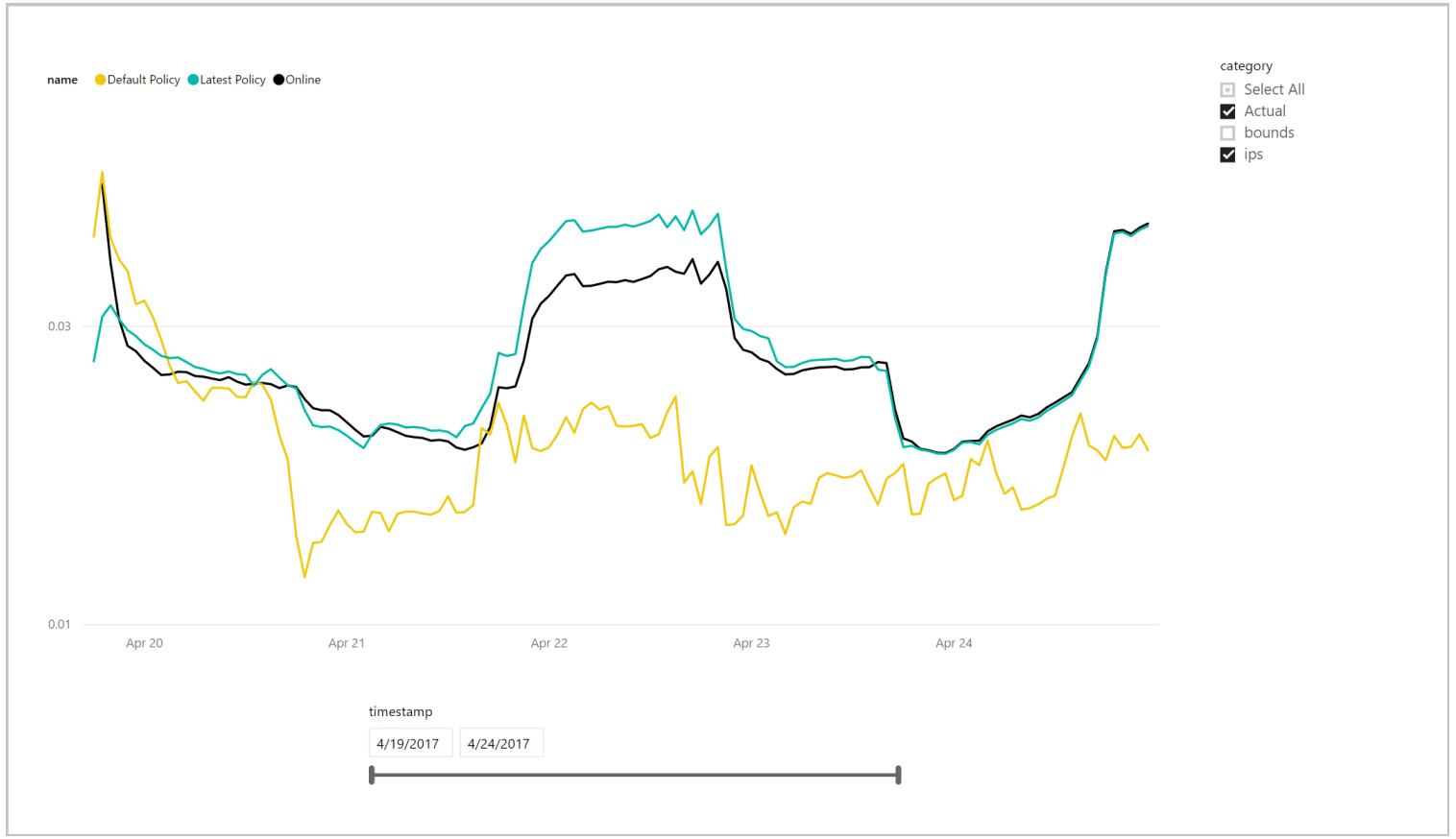
Self-host workflow in your Azure subscription

Offline Evaluation & Dashboard

Online performance

Offline estimate

- First article in RSS
- Latest model



Experimentation

https://github.com/Microsoft/mwt-ds/tree/master/DataScience

Complex Networks

POC overview

Jonathan Crockett
Director, R&D
e: jonathanc@complex.com

Complex Networks

The problem

Company

Complex Networks

A millennial focused, video first media company that is multi platform and reaches 57 million unique monthly visitors, 450 million monthly page views, and 300 million monthly video views.

Context

Pivoting towards video in an increasingly distributed platform landscape:

Facebook, Web, Twitter, YouTube, Instagram, Snapchat, [etc.], [etc.]

Problem statement

Users are increasingly consuming content off the web platform via ML generated recommendations.

How do we become smarter about surfacing the right content to the right people on the web?

The challenges

Challenge 1

Conversions

With mobile dominance, it's no longer a viable strategy to simply give the user more choices on one screen.

Surface the right piece content, and only the right piece of content.

Challenge 2

Responsiveness

The news cycle is now counted in minutes, not hours.

A piece of content can become high performing while the curators are out to lunch.

Challenge 3

Experimentation

How do we apply new methodologies in a way that can provide learnings and lessons to be applied to future experimentation and optimization?

How do we integrate with existing apps so it doesn't become a bottleneck?

How we employed the Decision Service

Identified use cases for experimentation. Article Video Widget and Homepage Secondary Hero slots.

POC with A/B testing.

Experimented tweaking features and configurations.

Continued A/B test against legacy solutions and other ML services. Reviewed results regularly and made adjustments over months.

Sync with other teams to review results and apply to new areas.

Tips

Identify a simple use case with a clear “reward”.

Employ A/B testing to experiment with featurization to verify success.

Use “conversion rate” for a metric to overcome browser inconsistency. Conversion rate defined as “If the user triggered the reward at least once per session”

Think of ML as an investment and not a magic bullet.

Start simple and iterate to become comfortable with applying ML.

More tips

Make sure you have a sterile testing environment.

Don’t discount the value of experimentation and learning phase.

Choose “action sets” wisely. The DS will only optimize relative to the choices you provide.

Have good metrics to observe user behavior for insights.

Have enough “experiments” to fuel model learning.

Areas for potential DS application

External performance

- Conversion Rates
- Sessions Time
- Revenue Generation

Internal efficiency.

- Editorial decision making
- Monitoring for trends

General Takeaways

Machine Learning is here to stay. ML has actually been here for a long time, just not in media companies.

The Decision Service is a great solution for media companies looking to explore the ML space and integrate deeply into their applications. It is low friction to use.

The best practices on applying ML including the Decision Service in a media company have not yet been codified. Every context is different.

ML is a tool that requires attention and skill to apply effectively.

Decision Ingredients

Shared features	Which actions?	Action dependent features
user history location browser	articles videos	celebrities sentiment categories

What's a good application?

Many decisions	End-user interaction, not back office
Response ↔ Decision	Link click to ranking
Available actions <20	Per decision, total pool can be large

Custom Decision Service

A cloud-based, contextual decision-making API that sharpens with experience

Contextual

Understanding context from information you provide.
Ranks the options and makes a decision.

Rapid learning

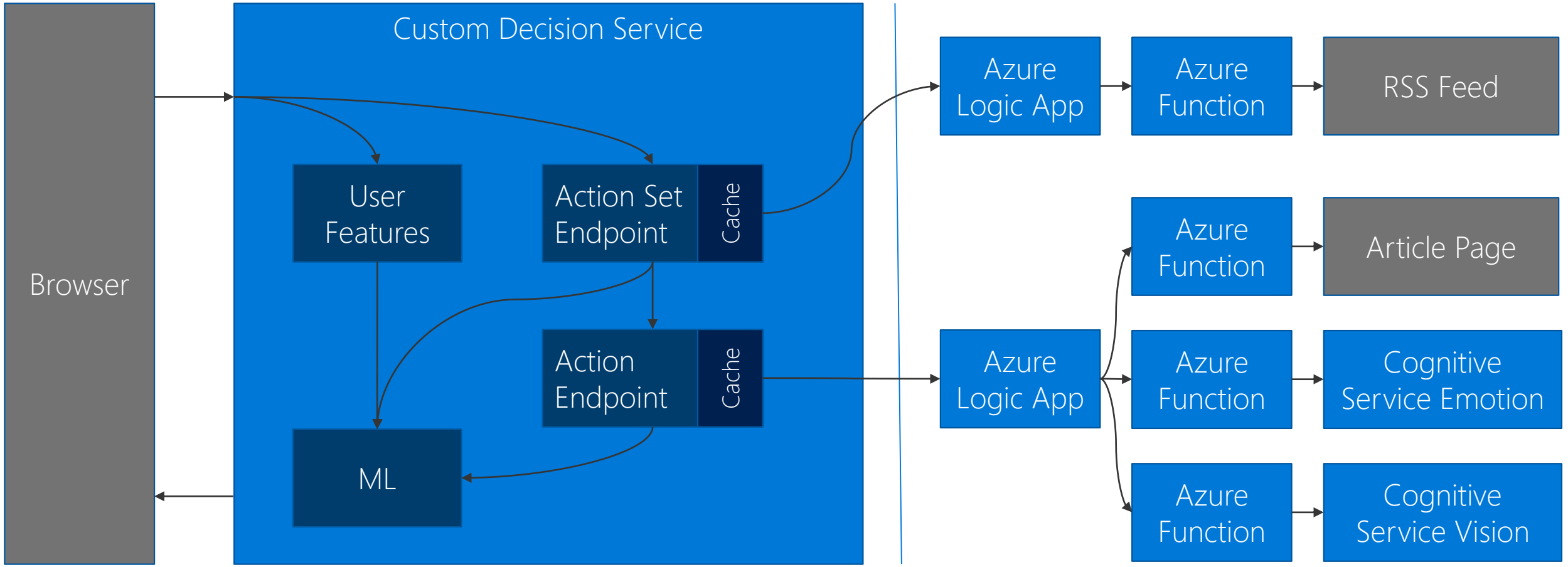
Custom Decision Service automatically optimizes based on your feedback. It constantly explores new options to see if the best decision has changed, enabling it to adapt to emerging trends.

Easy to use

Custom Decision Service is cloud-based so it's easy to run, able to plug into your application and help to make decisions in real time.

Thank you!

Customize Auto-featurization



Easy: RSS feed + Javascript

Microsoft

Custom Decision Service Demo

Usage

Step 1: Embed script in page with callback function:

```
<script src="//ds.microsoft.com/api/v2/demo/rank/recent?details=2" async>
</script>
```

```
function callback(json) {
    $.map(json.ranking, function (element) {
        //Custom rendering function given content info
        render({img:element.details[1].image,
                title:element.details[1].title,
                onClick: element.id != json.rewardAction ? null :
                   function() {
                       $.ajax({
                       type: "POST",
                       url: '//ds.microsoft.com/api/v2/demo/reward/' + jso
n.eventId,
                       contentType: "application/json"
                   }}
                });
}
```

Click on the most interesting article

Growing consensus on the
state attacks - Microsoft on

Satya Nadella to employee
Official Microsoft Blog