Game Analytics & Machine Learning

Guest Lecture: UCSC CMPM 146

expressiveintelligencestudio

Ben Weber

Sr. Data Scientist

Electronic Arts

beweber@ea.com

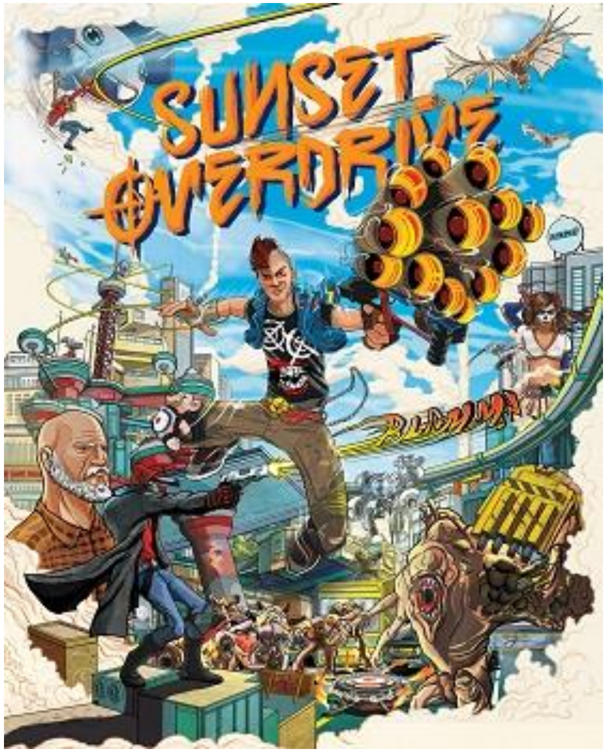
GAMES AND PLAYABLE MEDIA

UC SANTA CRUZ

About Me

- **Ben G. Weber**
  - Senior Data Scientist
  - Electronic Arts
- **Experience**
  - PhD in Computer Science, UCSC
  - Technical Analyst Intern, EA
  - User Research Analyst, Microsoft Studios
  - Ecommerce Analyst, Sony Online Entertainment
  - Director of BI & Analytics, Daybreak Games

Games & Services



Origin™ access

Talk Overview

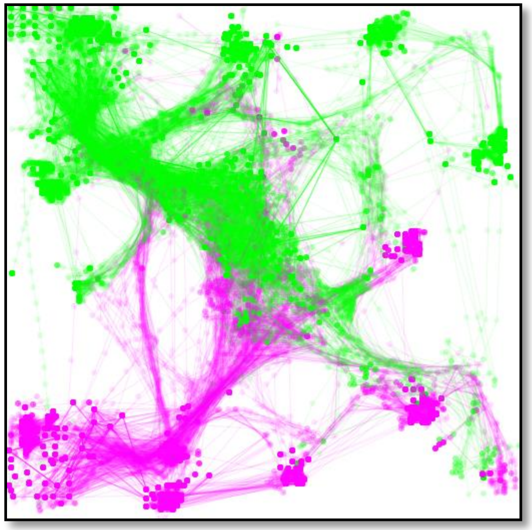
- **Game Analytics**
- **Classification Algorithms**
  - Strategy Prediction in StarCraft: Brood War
  - Membership Conversion in DC Universe Online
- **Regression Algorithms**
  - Player Retention in Madden NFL
- **Recommendation Algorithms**
  - Recommendation System in EverQuest Landmark
- **Data Science at EA**

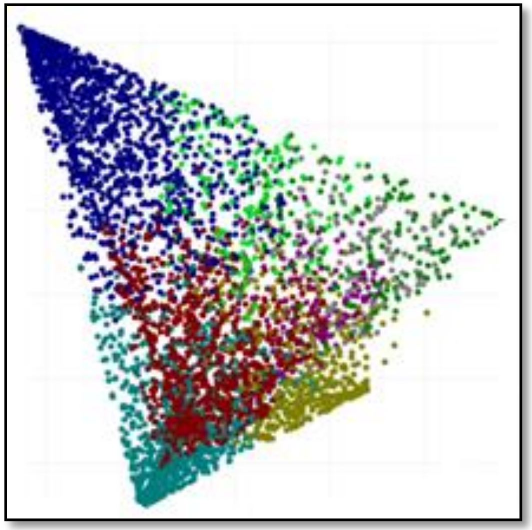
What is Game Analytics?

- Using data to inform game design and improve player lifecycles
- Gathering feedback from players, making changes, and measuring the impact
- **Example Applications**
  - Tuning game difficulty
  - Balancing the game economy
  - Content recommendations
  - Computing the lifetime value of players

What Data Can We Analyze?

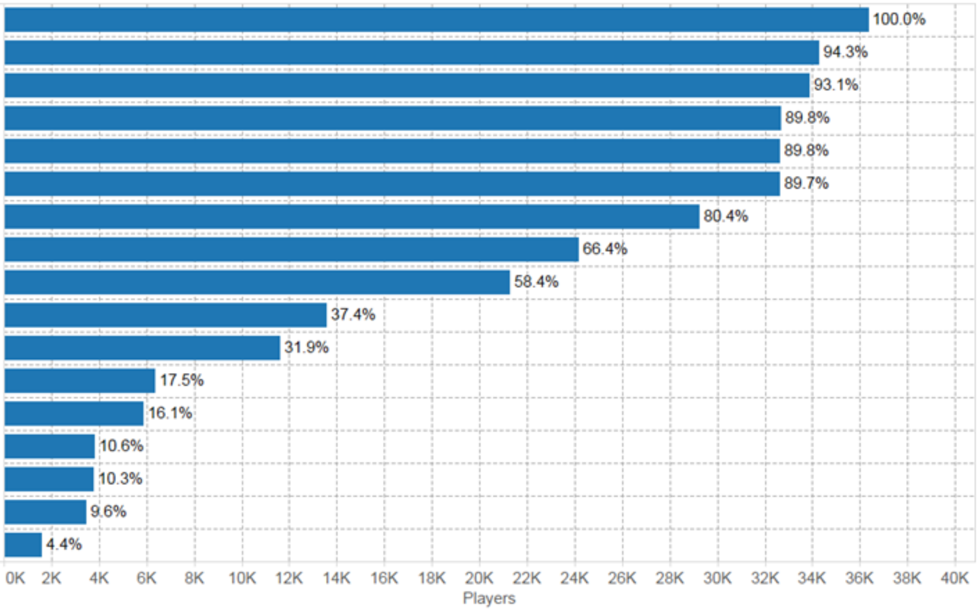
- **Game data**
  - In-Game events
  - Commerce events
  - Operational metrics
- **User Data**
  - Web sessions
  - Acquisition channel
- **3rd Party Data**
  - Steam Spy
  - Platform Data





How can we analyze players?

- Descriptive statistics and exploratory data analysis
- Experimentation & Measurement
- Segmentation
- Forecasting
- Predictive analytics



What teams use Analytics?

- Design
- Product Management
- Marketing
- Business Development
- Strategy
- Finance
- Dev Ops
- Community Management

Analytics for Design

- **Question:** Identify questions about the current design
- **Record:** Enumerate which data needs to be collected and deploy the game
- **Analyze:** Determine if the recorded data matches expectations
- **Refine:** Given the findings, analyze the design and formulate additional questions

Tools for Analytics

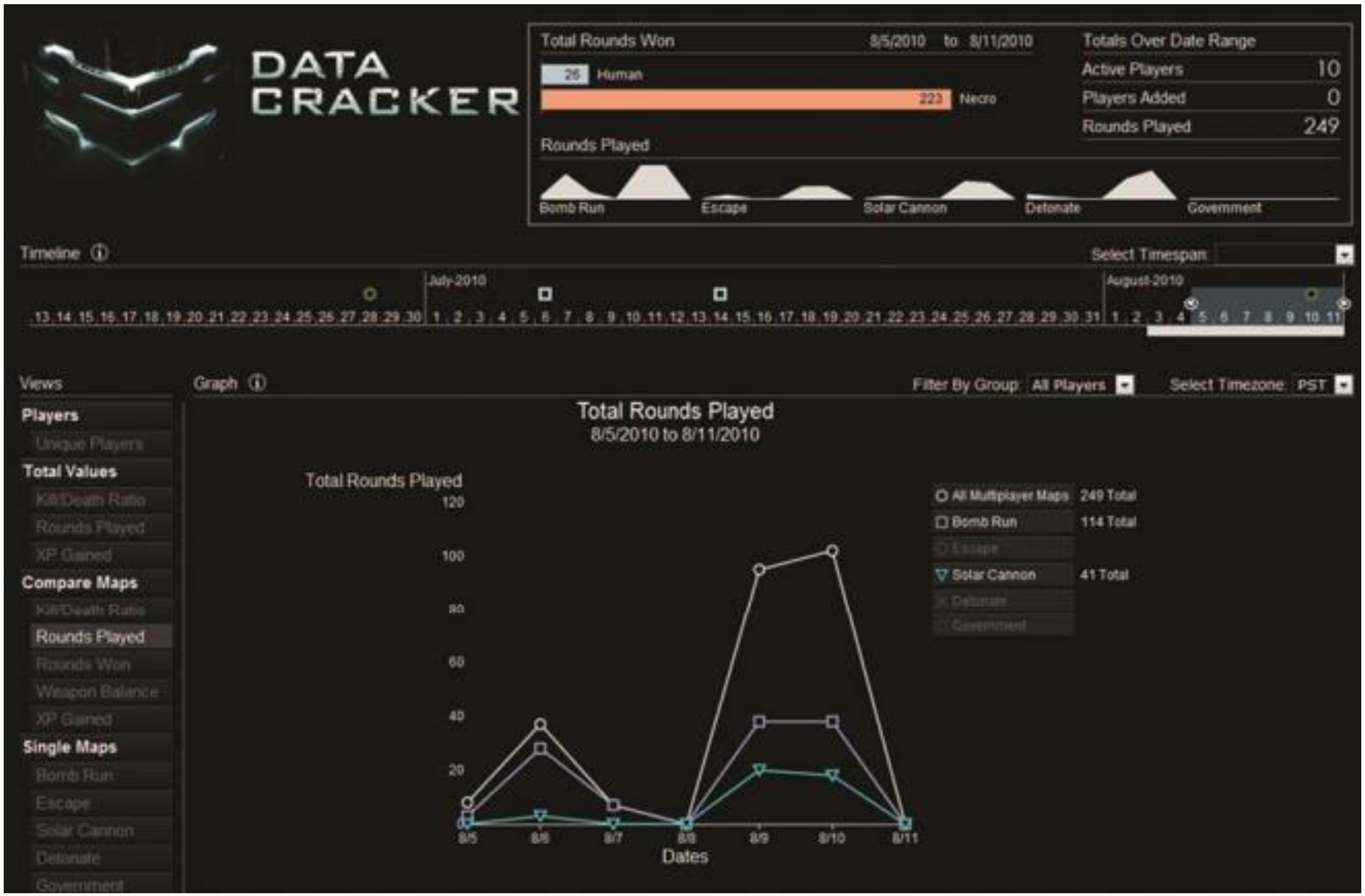
- **Storage**
  - DBMS, Hadoop, Cloud Storage
- **Reporting**
  - Excel, MicroStrategy, Tableau, D3
- **Analysis**
  - SQL, R, Python, Scala
- **Model Building**
  - R, Weka, MLlib

Applications of Analytics in Games

- Dashboard Analytics (Metrics & KPIs)
- Spatial Analytics (Visualization)
- User Research Analytics
- Market Research Analytics
- Data Science

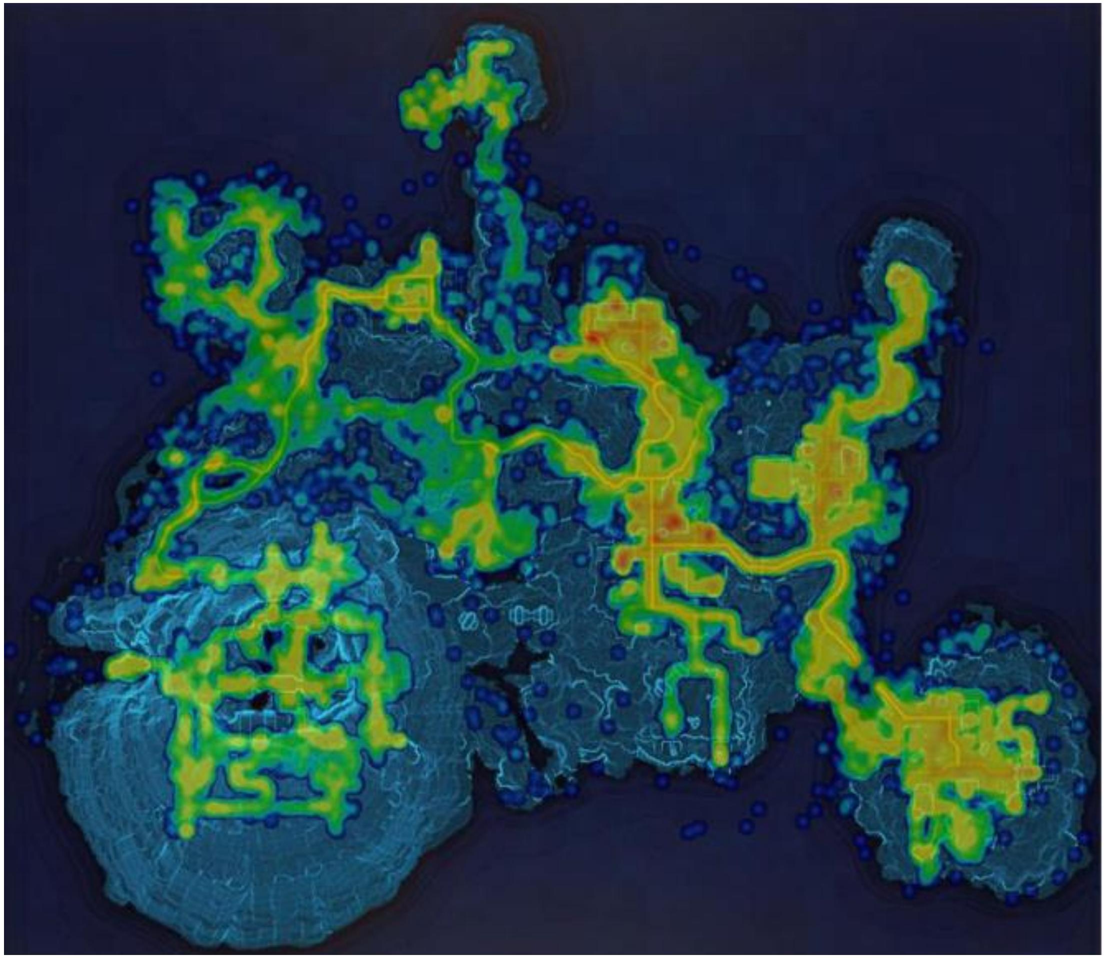
Dashboard Analytics

- Dead Space 2 Data Cracker (Ben Medler, GDC 2011)



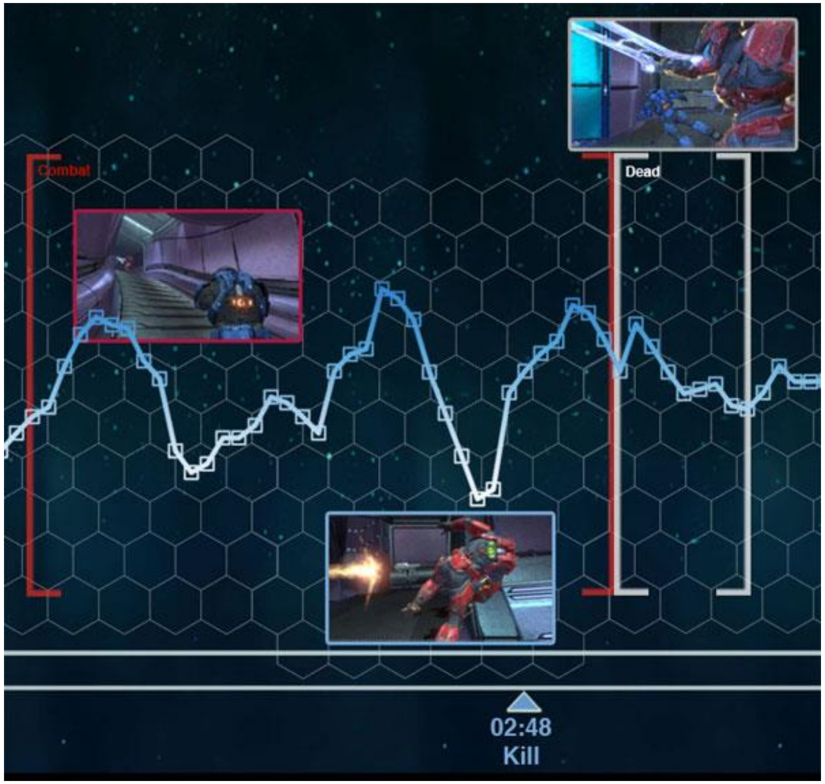
Spatial Analytics

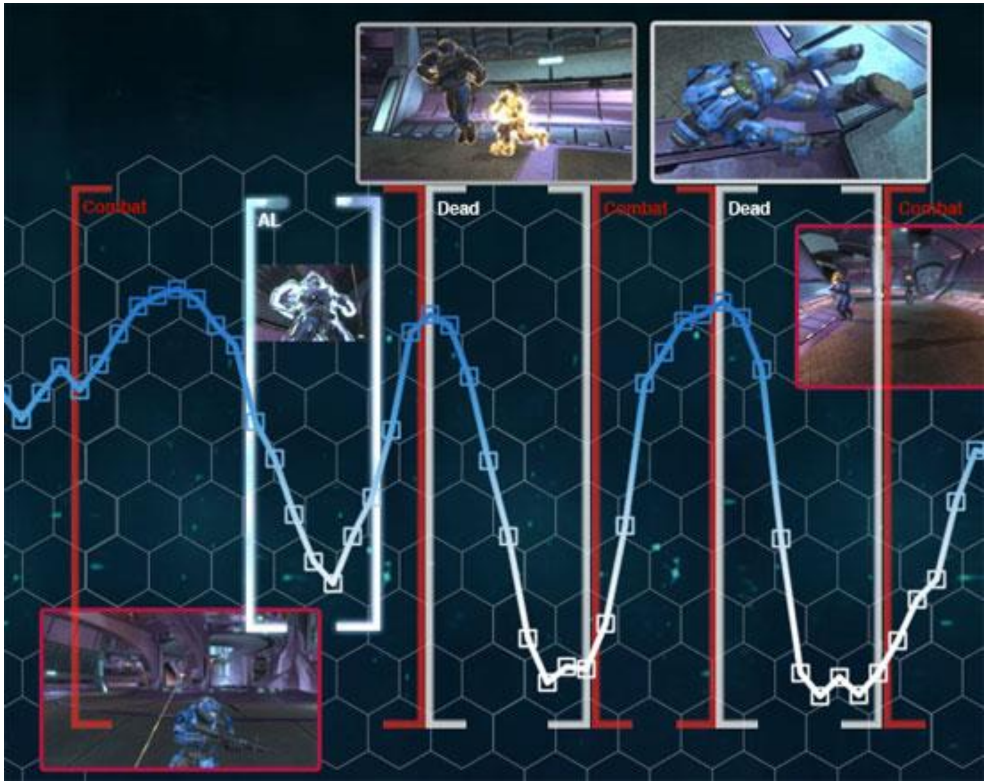
- Player movement in SWTOR (Georg Zoeller, GDC 2010)



User Research Analytics

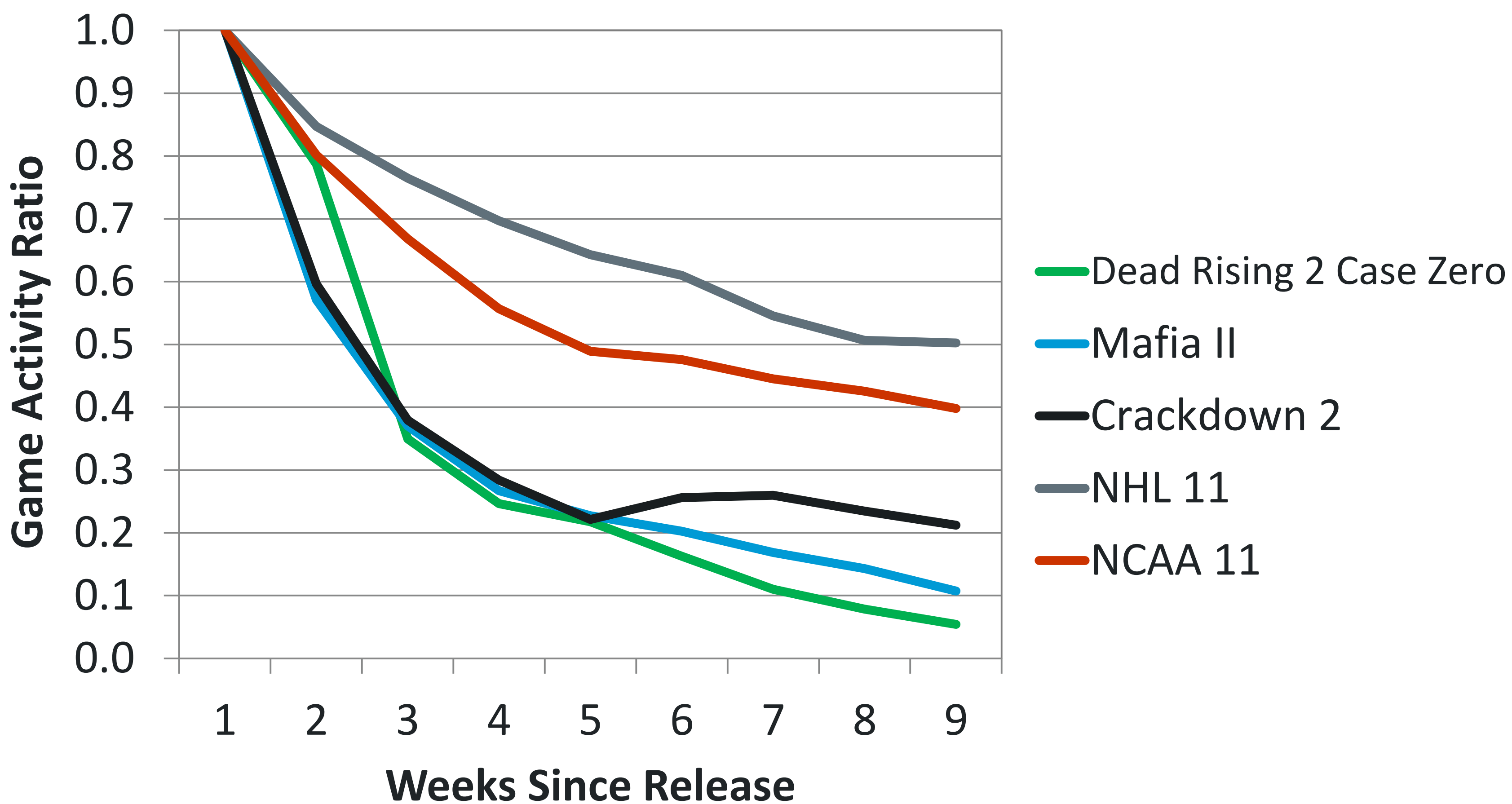
- My Heart on Halo, Ben Lewis-Evans





Market Research Analytics

- Player Retention data scraped from Raptr.com



Recommended Reading (Gamasutra)

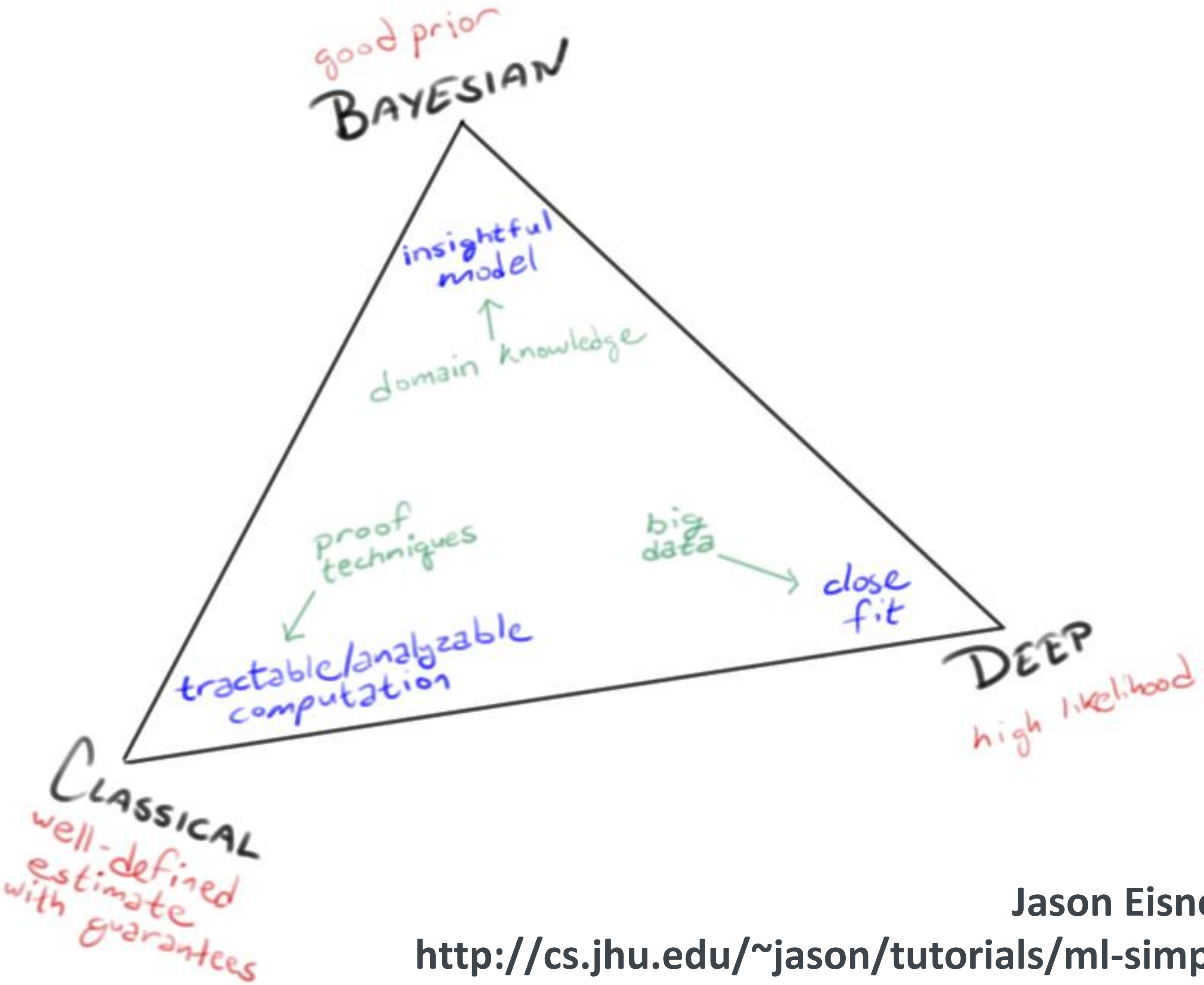
- **Intro to User Analytics**
  - What to track and how to analyze data
  - Anders Drachen *et al.*
- **Game Analytics 101**
  - What technologies to use?
  - Dmitri Williams
- **Indie Game Analytics 101**
  - What metrics to track?
  - Dylan Jones

Machine Learning (ML)

- Algorithms that learn from data and make inferences or predictions

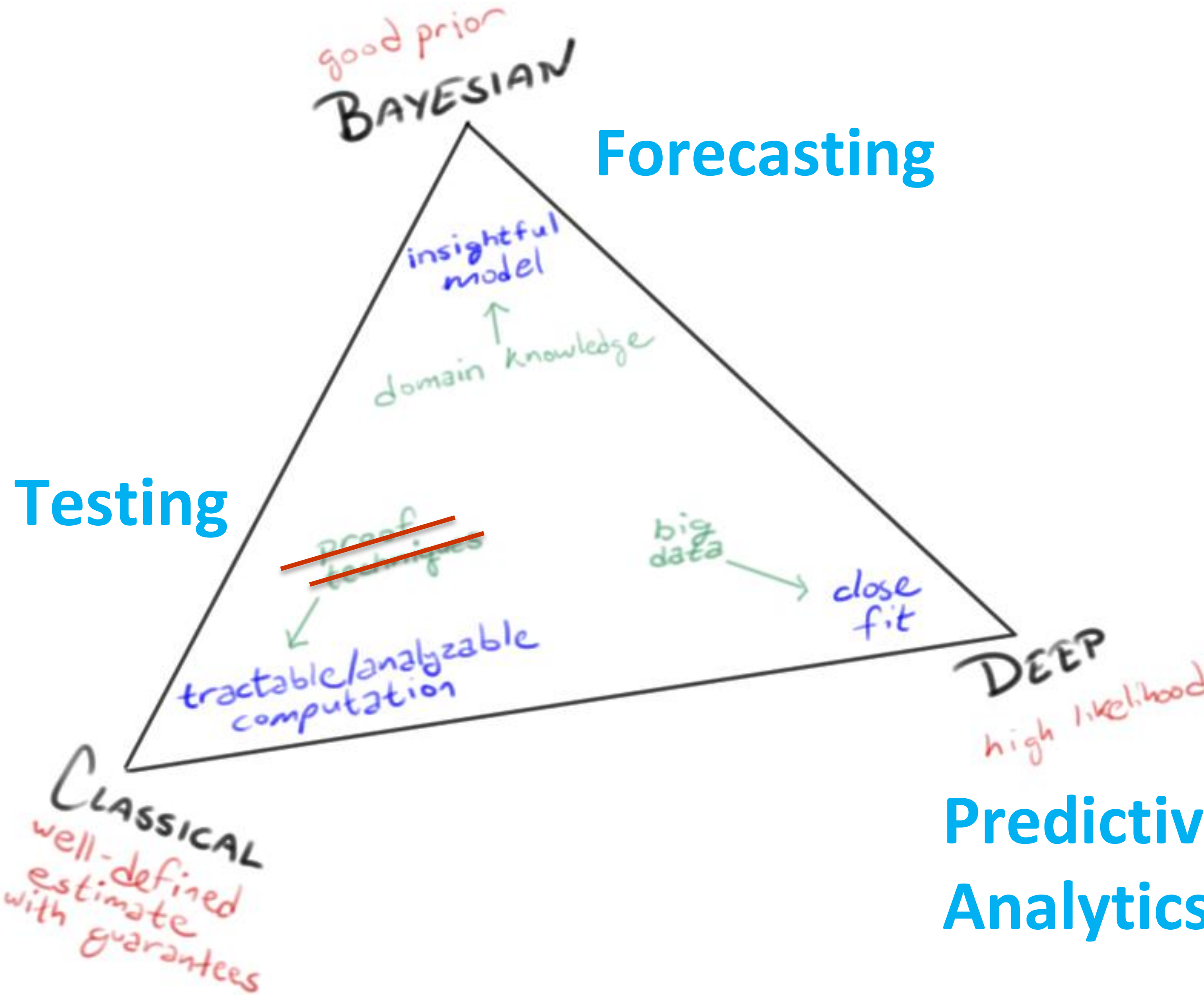
- **Types of Problems**
  - **Classification:**
    - Identifying the category of an instance
    - Prediction: label
  - **Regression:**
    - Estimating relationships between variables
    - Prediction: value
  - **Clustering**
    - Identifying similar groups of instances

The 3 Cultures of Machine Learning



Jason Eisner (2015)
http://cs.jhu.edu/~jason/tutorials/ml-simplex.html

Application to Game Analytics



Forecasting

A/B Testing

Predictive Analytics

Applying ML to Games

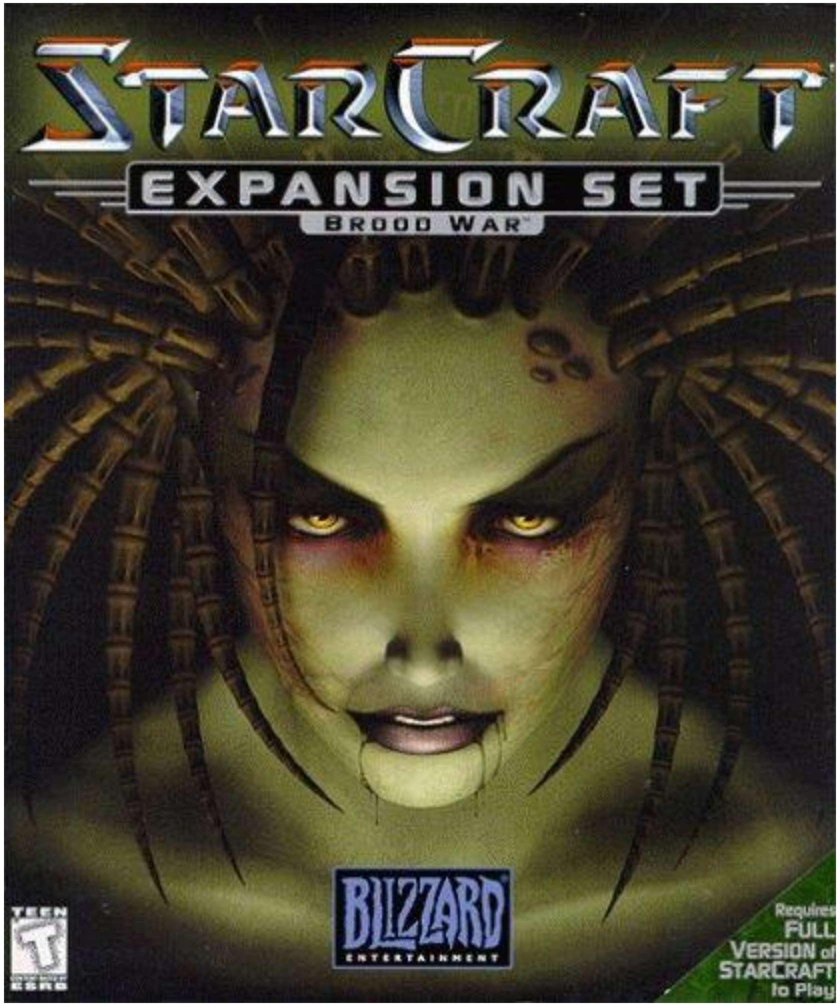
- How to **represent** the problem?
- What **features** to use?
- How to **evaluate** the **model** being produced?
- How to **deploy** the **model**?

Classification Algorithms

- **Goal**
  - Identify the category that an instance belongs to
- **Examples**
  - Is a player going to make a purchase?
  - What is an opponent's build order?
  - Is a user going to quit playing this week?
- **Algorithms**
  - Decision Trees
  - Logistic Regression
  - Neural Networks
  - Boosting
  - Support Vector Machines
  - Nearest Neighbor

Strategy Prediction in StarCraft

StarCraft: Brood War, Blizzard Entertainment 1998



Predicting Build-Orders in StarCraft

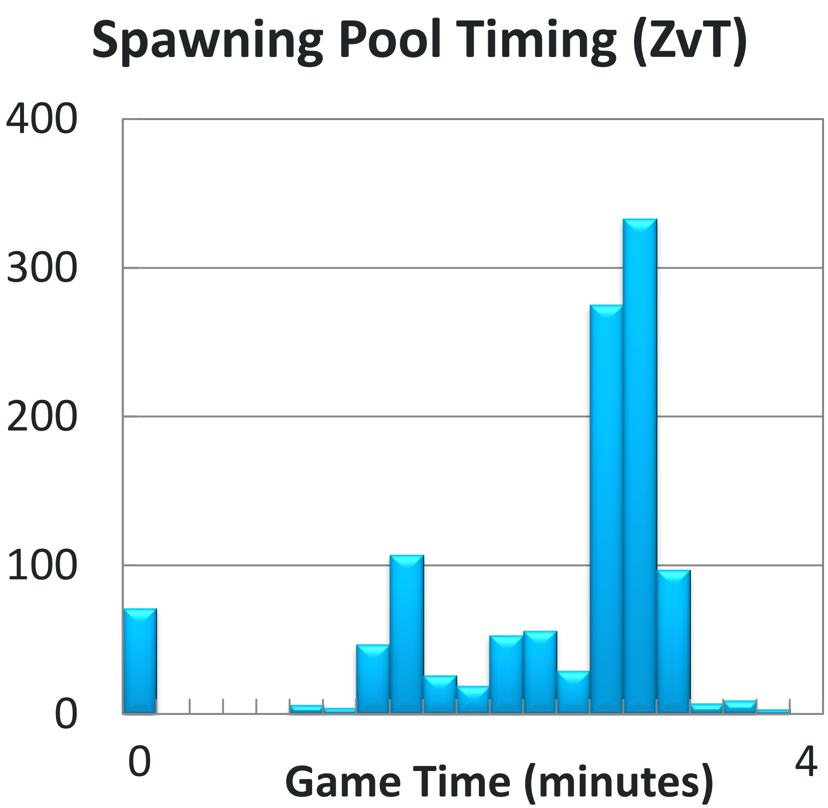
- **Goal**
  - Predict the build order of opponents
- **Data Sources**
  - Thousands of replays from Professional players
- **Results**
  - Able to identify opponent build order minutes before it is executed
  - Also able to predict the timing of specific units

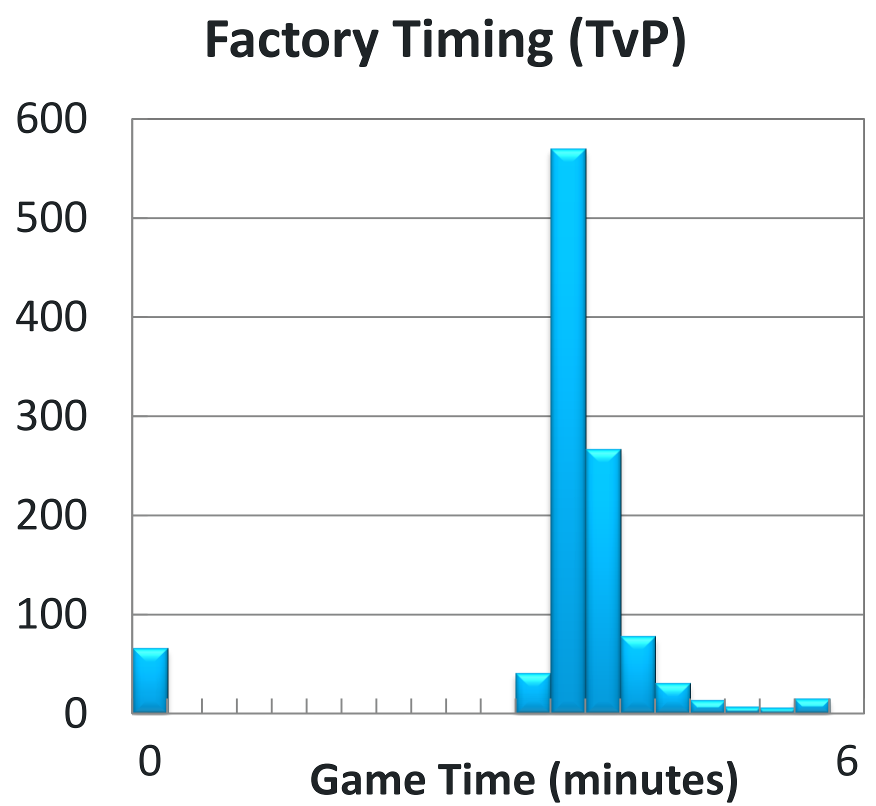
Application of ML to StarCraft

- **Problem Representation**
  - Multiclass classification, 6 labels for mid-game builds
  - Collect data from professional StarCraft replays
- **Features**
  - Build-order timings of units and features
- **Model Evaluation**
  - Offline evaluation of classification algorithms
  - Simulated game time between 0 – 12 minutes
- **Model Deployment**
  - During games, encode game state and predict build order

Timing Distributions

- Different structure and units timings provide indicators of different build orders





StarCraft Replay Data

- A partial game log from a Terran versus Zerg game

Player	Game Time	Action
2	0:00	Train Drone
1	0:00	Train SCV
2	1:18	Train Overlord
1	1:22	Build Supply Depot
1	2:04	Build Barracks
2	2:25	Build Hatchery
1	2:50	Build Barracks
2	2:54	Build Spawning Pool
1	3:18	Train Marine
2	4:10	Train Zergling

Approach

- **Feature encoding**
  - Each player's actions are encoded in a single vector
  - Vectors are labeled using a build-order rule set

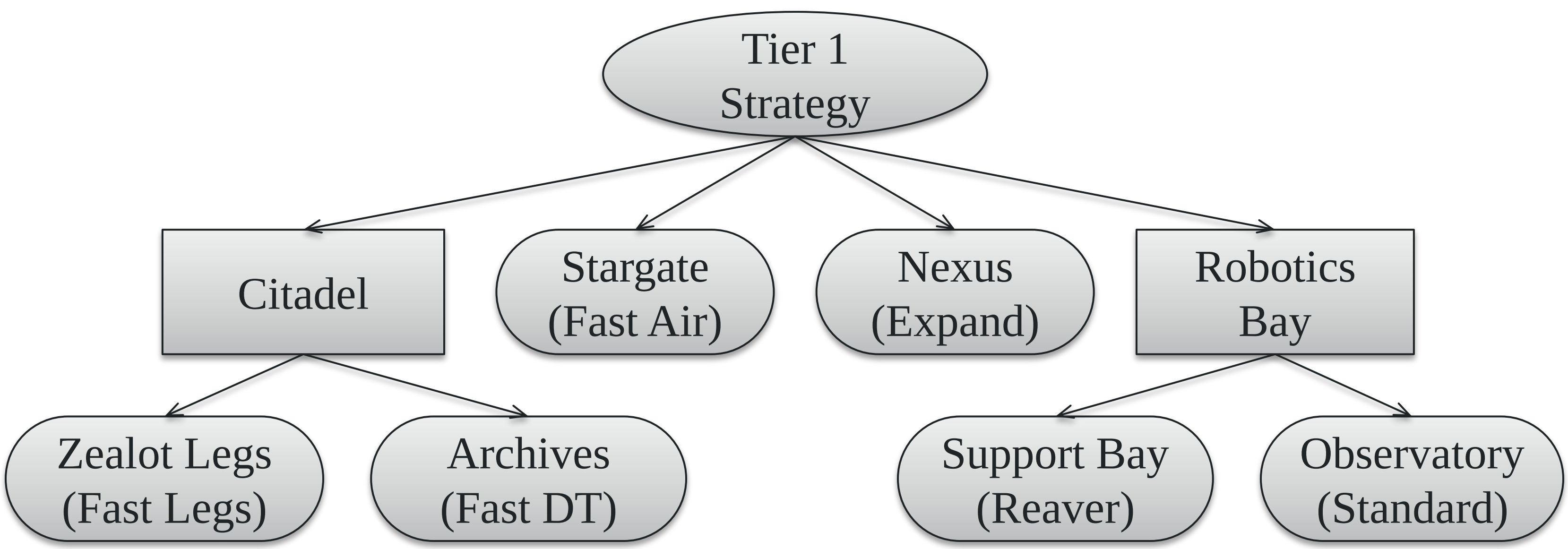
- Features describe the game cycle when a unit or building type is first produced by a player

$$f(x) = \begin{cases} t, & \text{time when } \mathbf{x} \text{ is first produced by } \boldsymbol{P} \\ 0, & \boldsymbol{x} \text{ was not (yet) produced by } \boldsymbol{P} \end{cases}$$

Labeling Replays

- A rule set for mid game strategies was built for each race based on analysis of expert play
- Replays are labeled based on the order in which the tech tree is expanded

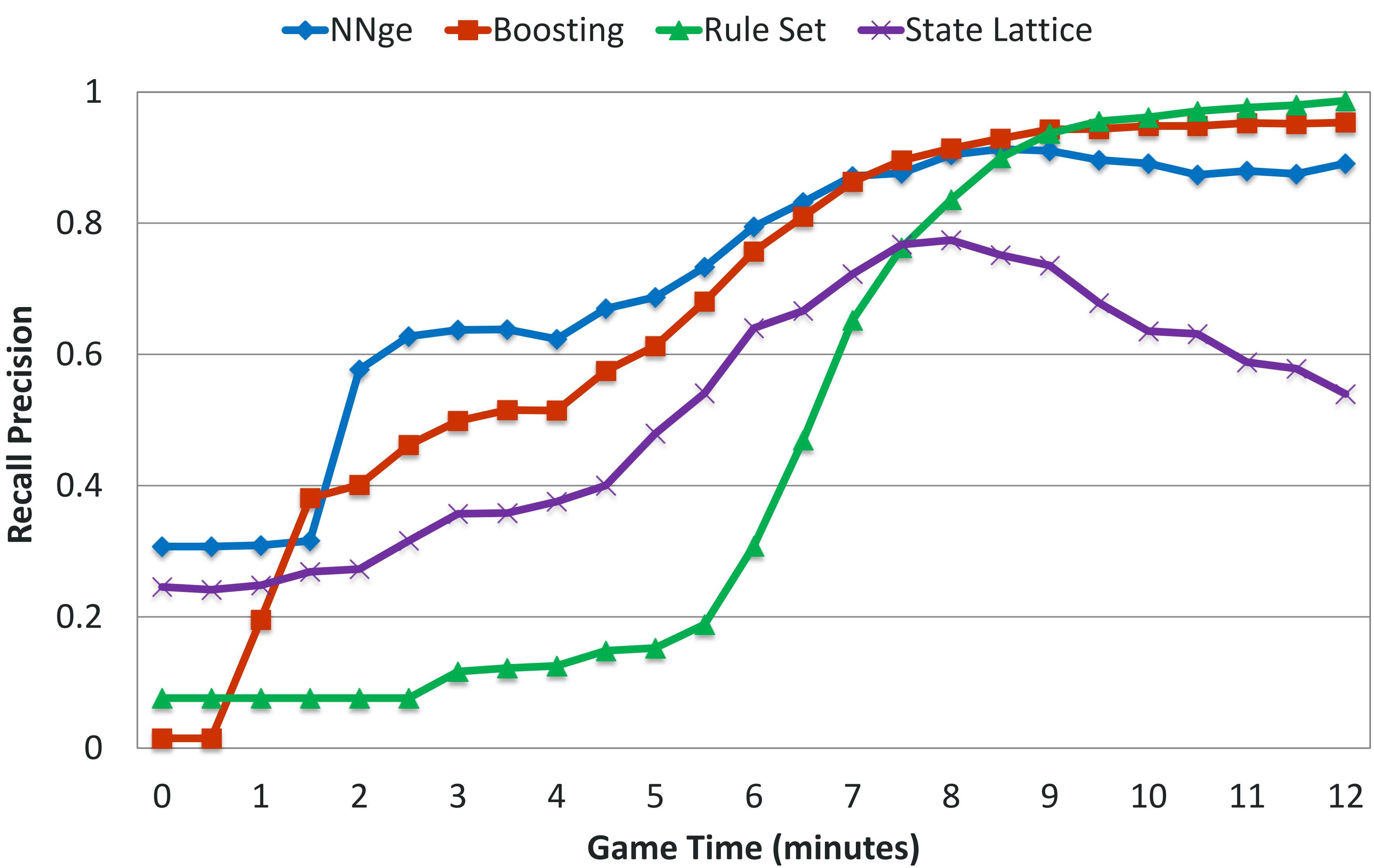
Protoss Rule Set



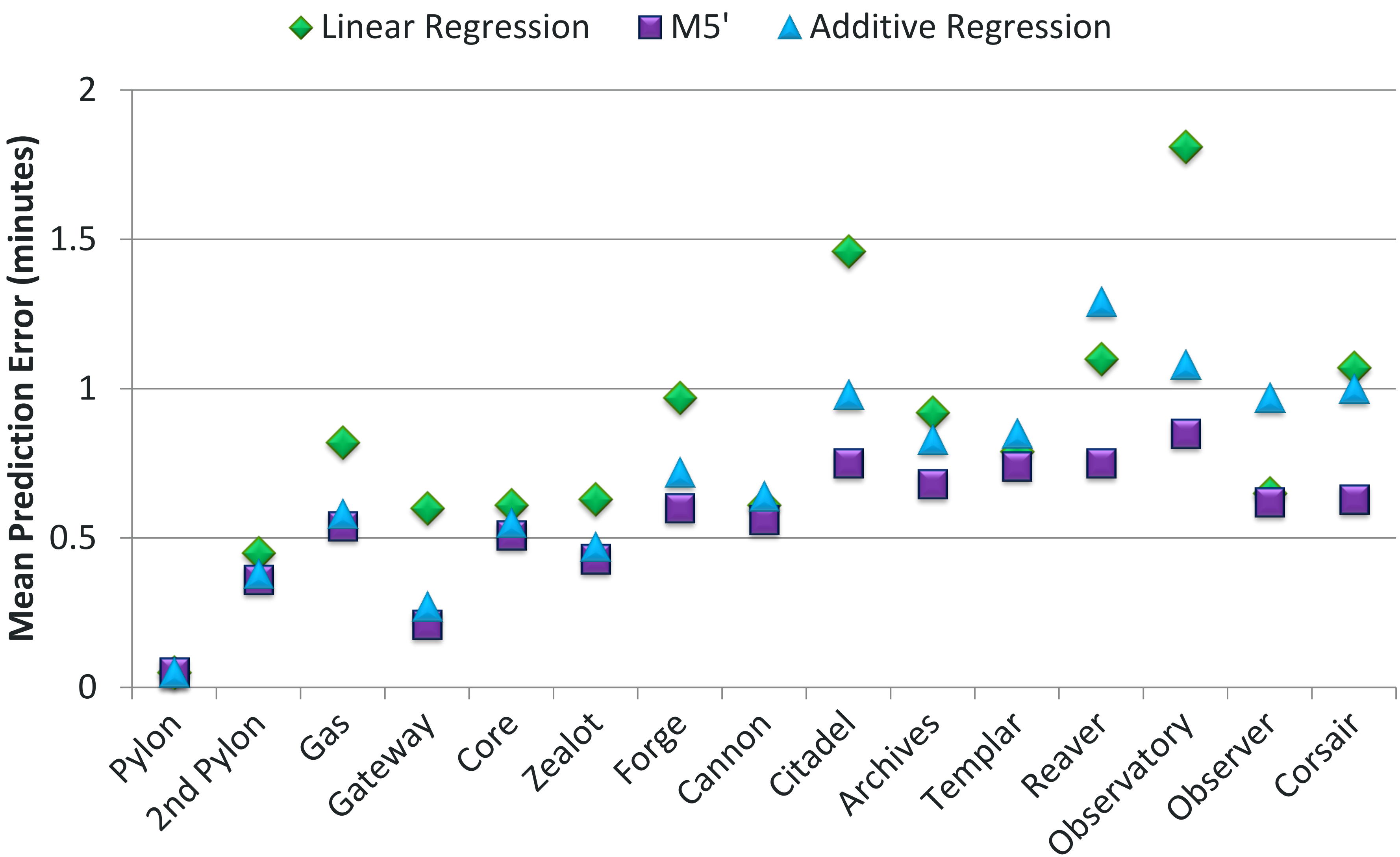
Experiment Methodology

- **Algorithms explored**
  - Nearest neighbor variants
  - Decision trees
  - Boosting methods
  - State lattice
  - Rule set
- **Simulation Approach**
  - Set all features to 0
  - Step through replay events and update features
  - Predict build order every 30 seconds

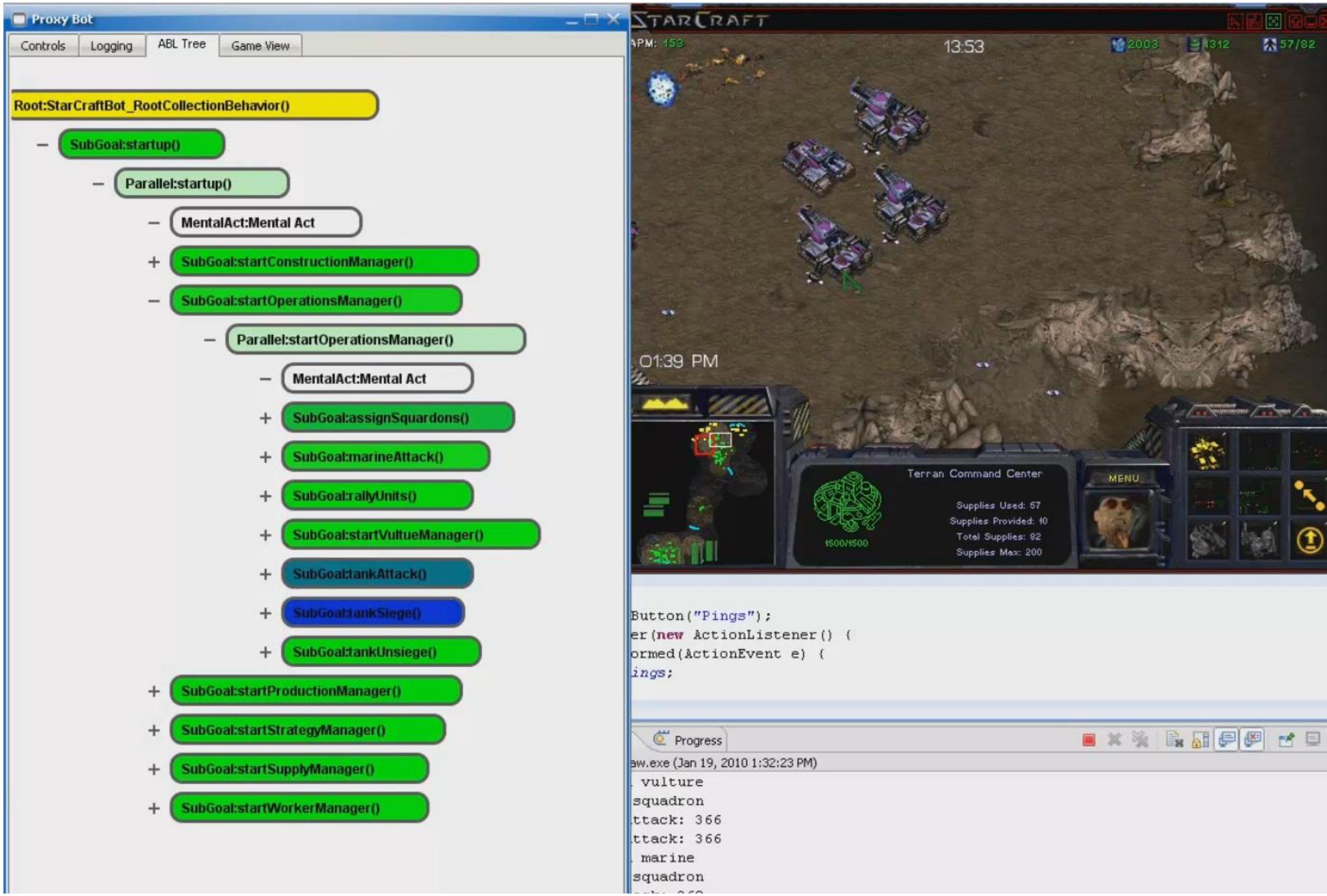
Build-Order Prediction Results



Timing Prediction Results



EISBot



Proxy Bot
Controls
Logging
ABL Tree
Game View
Root:StarCraftBot_RootCollectionBehavior()
SubGoal:startup()
Parallel:startup()
MentalAct:Mental Act
SubGoal:startConstructionManager()
SubGoal:startOperationsManager()
Parallel:startOperationsManager()
MentalAct:Mental Act
SubGoal:assignSquardons()
SubGoal:marineAttack()
SubGoal:rallyUnits()
SubGoal:startVultueManager()
SubGoal:tankAttack()
SubGoal:tankSiege()
SubGoal:tankUnsiege()
SubGoal:startProductionManager()
SubGoal:startStrategyManager()
SubGoal:startSupplyManager()
SubGoal:startWorkerManager()
STARCRAFT
APM: 153
13:53
2003
1312
57/82
01:39 PM
Terran Command Center
Supplies Used: 57
Supplies Provided: 10
Total Supplies: 82
Supplies Max: 200
1500/1500
MENU
Button("Pings");
er(new ActionListener() {
ormed(ActionEvent e) {
ings;
Progress
aw.exe (Jan 19, 2010 1:32:23 PM)
vulture
squadron
ttack: 366
ttack: 366
marine
squadron

Membership Conversion in DCUO

DC Universe Online, Daybreak Games 2011

Membership Conversion in DCUO

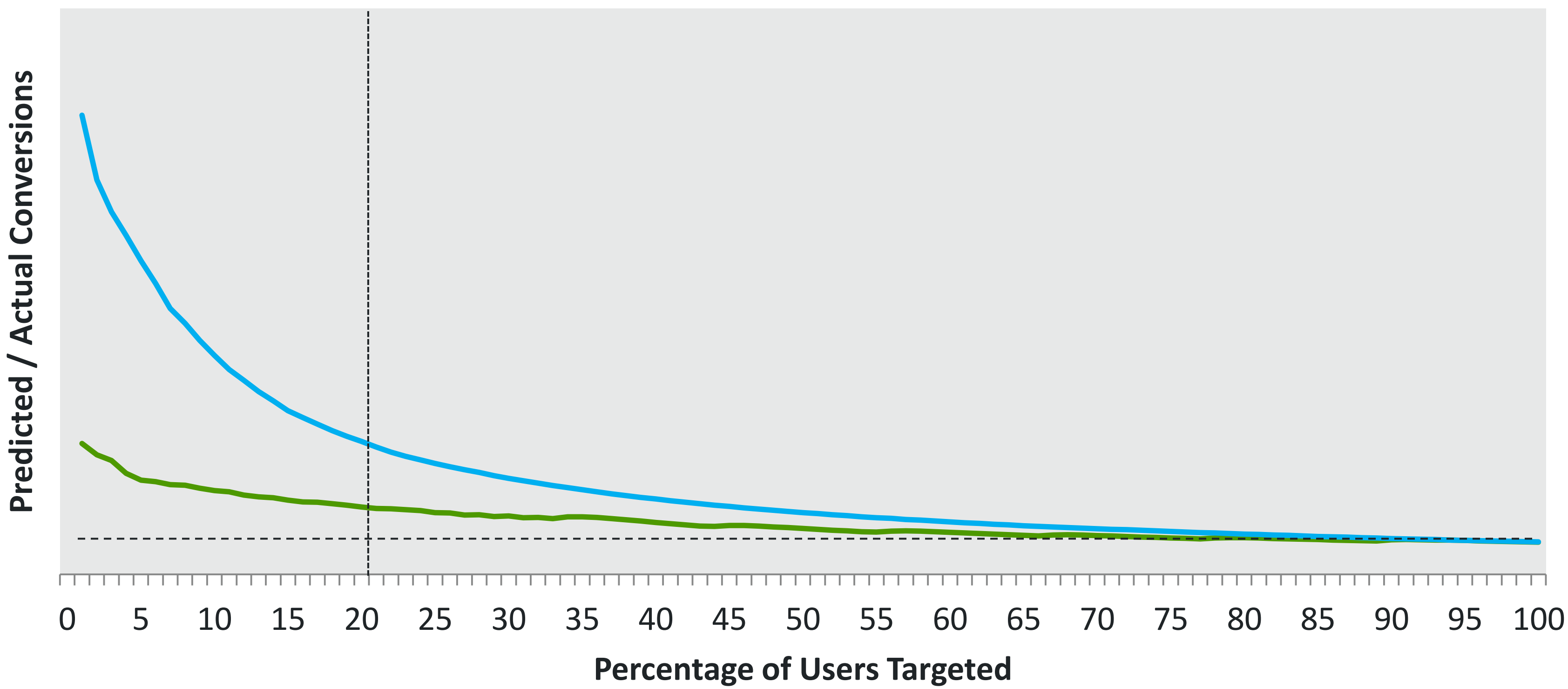
- **Goal**
  - Predict which users will convert to membership
- **Data Sources**
  - Session and commerce data
  - Detailed in-game telemetry
- **Results**
  - Weekly deployment of targeted user list
  - Experimented with different targeting strategies and uplift modeling

Application of ML to DCUO

- **Problem Representation**
  - Binary classification (Converter vs. non-converter)
- **Features**
  - Login patterns, recent purchases, game-feature usage
- **Model Evaluation**
  - Offline evaluation of classification algorithms
  - Simulated upsell
- **Model Deployment**
  - Send list of targeted users to game team

Model Evaluation

- **Lift**
  - Compares sampled response to baseline response



Uplift Modeling

- **Goal**
  - Target only persuadable users

Source: Predictive Analytics (Eric Siegel, 2013)

Uplift Modeling in DCUO

- **2-Phase Approach**
- **Phase 1**
  - Select users for targeting
- **Phase 2**
  - Split users into Sure Things & Persuadable
  - Target users in the Persuadable group

Measuring Impact

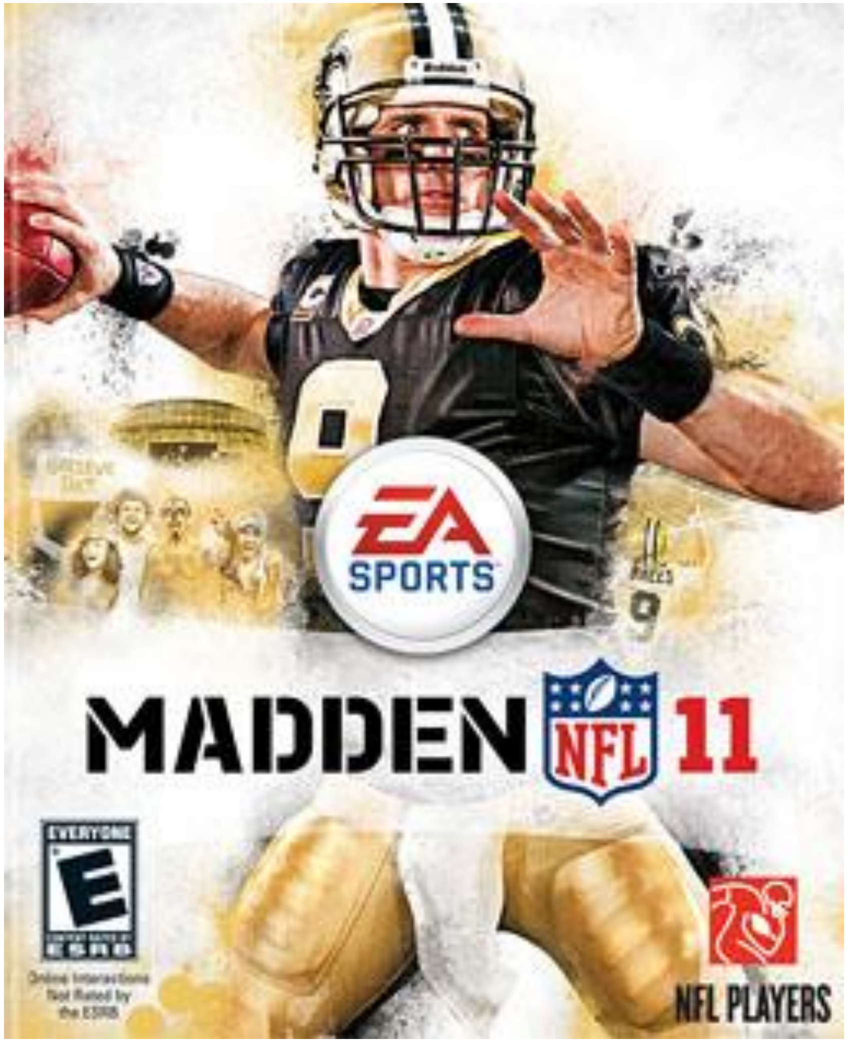
- **Goal**
  - Maximize Incremental revenue
- **Approach**
  - Select targeted user group
  - Split targeted group into control and test
  - Deploy targeting and measure conversions
  - Compare test and control groups
  - Calculate incremental revenue

Regression Algorithms

- **Goal**
  - Predict the value of the dependent variable of an instance
- **Examples**
  - How many games is a user going to play?
  - What is the lifetime value of a player?
  - How to allocate players for balanced matchmaking?
- **Algorithms**
  - Linear Regression
  - Regression Tree
  - Curve fitting
  - Boosting
  - Neural Networks
  - Nearest Neighbor

Player Retention in Madden NFL

Madden NFL 11, Electronic Arts 2010



Player Retention in Madden NFL

- **Goal**
  - Predict how long players will play
  - Identify features correlated with retention
- **Data Sources**
  - Play-by-play game logs
- **Results**
  - Recommendations provided to game team
  - Helped develop data-driven culture at EA

Application of ML to Madden

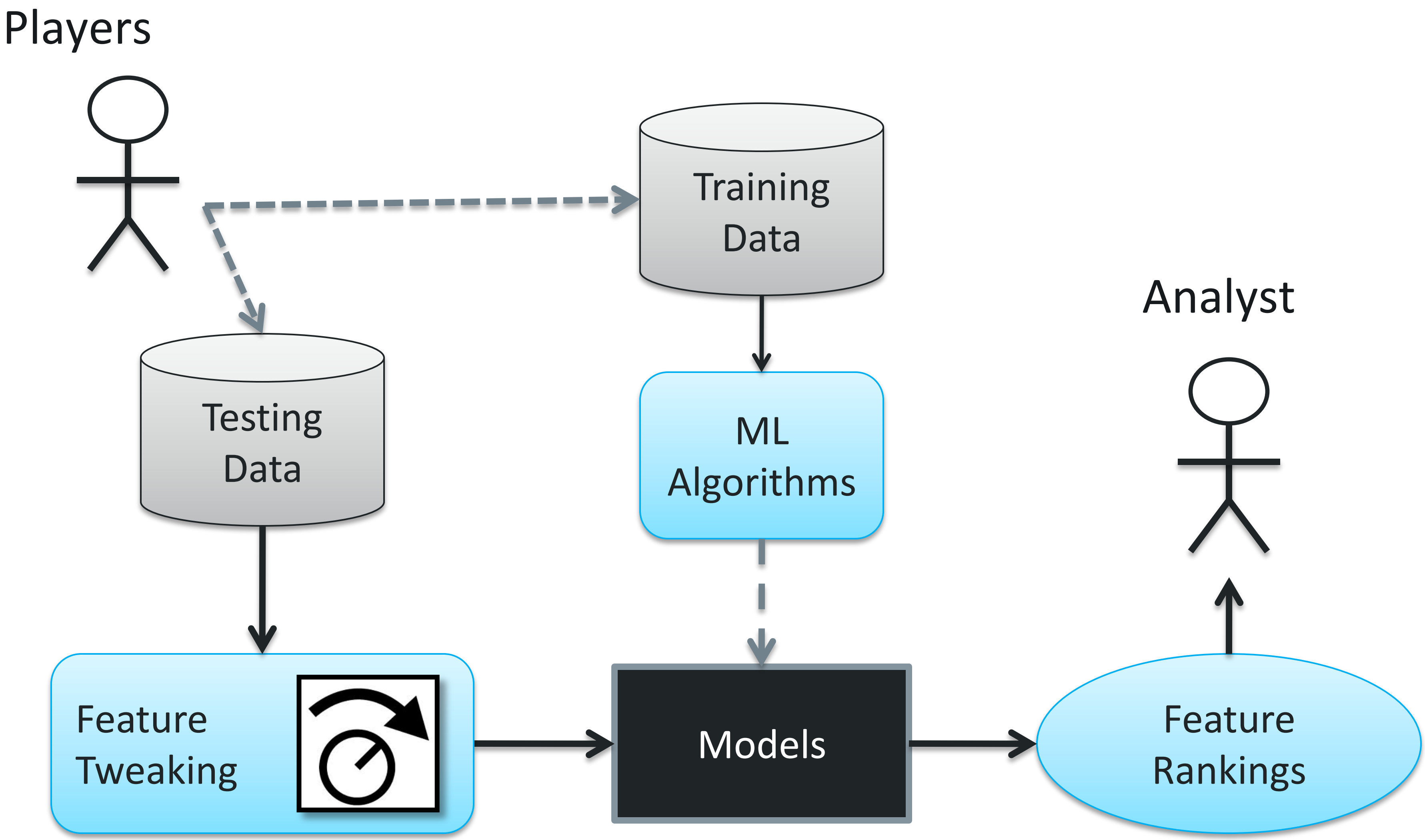
- **Problem Representation**
  - Regression & Simulation
- **Features**
  - Gameplay features used
  - Player performance
- **Model Evaluation**
  - Offline evaluation of regression algorithms
  - Unique Effect Analysis
- **Model Deployment**
  - Recommendations provided to game team

Robust Unique Effect Analysis

- An algorithm that performs **regression** and analyzes **unique effects** to rank features

- **Algorithm overview**
  1. Build regression models for predicting retention
  2. Perturb the inputs to the models
  3. Compute the impact of individual features

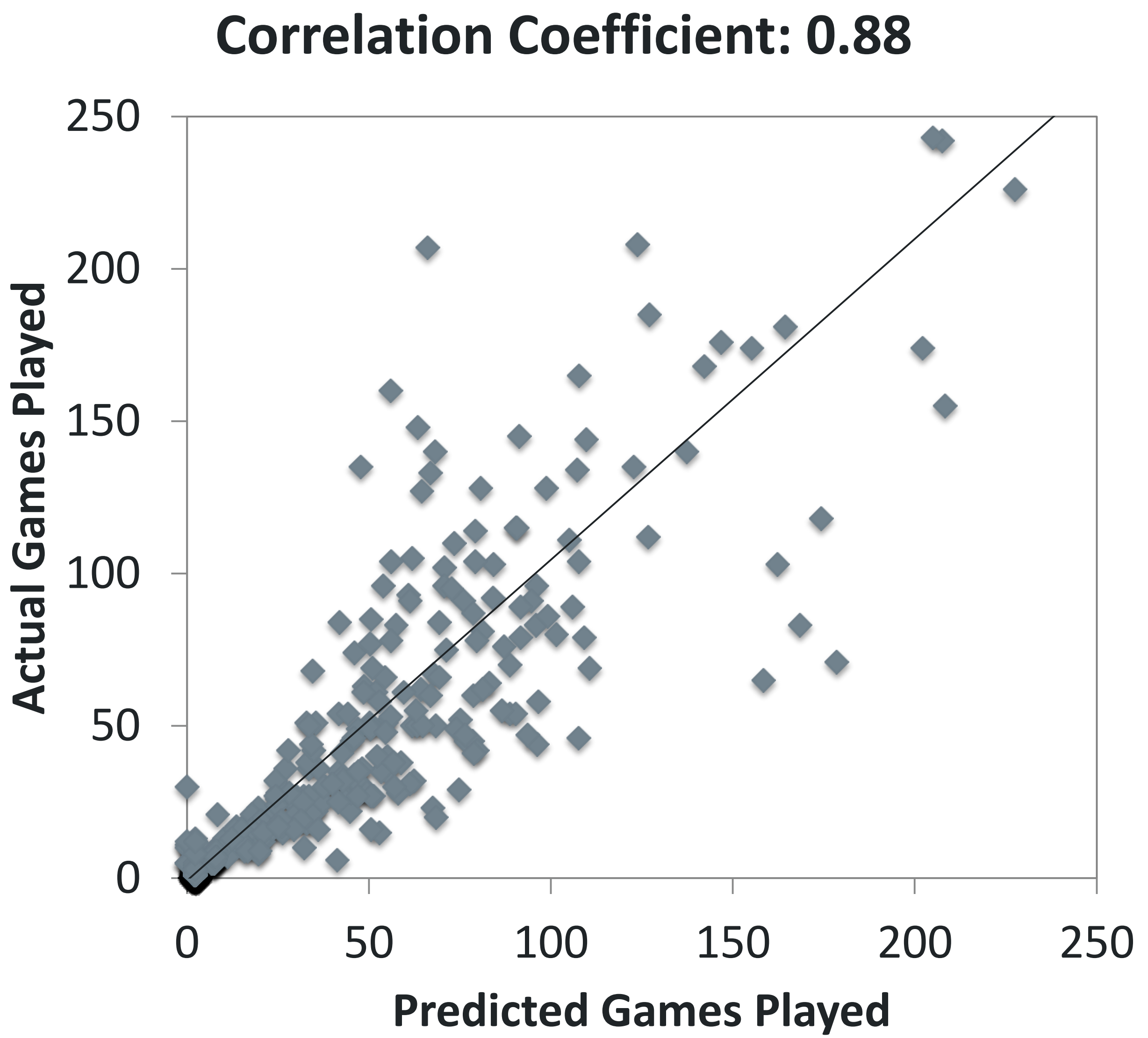
Algorithm Overview



Player Representation

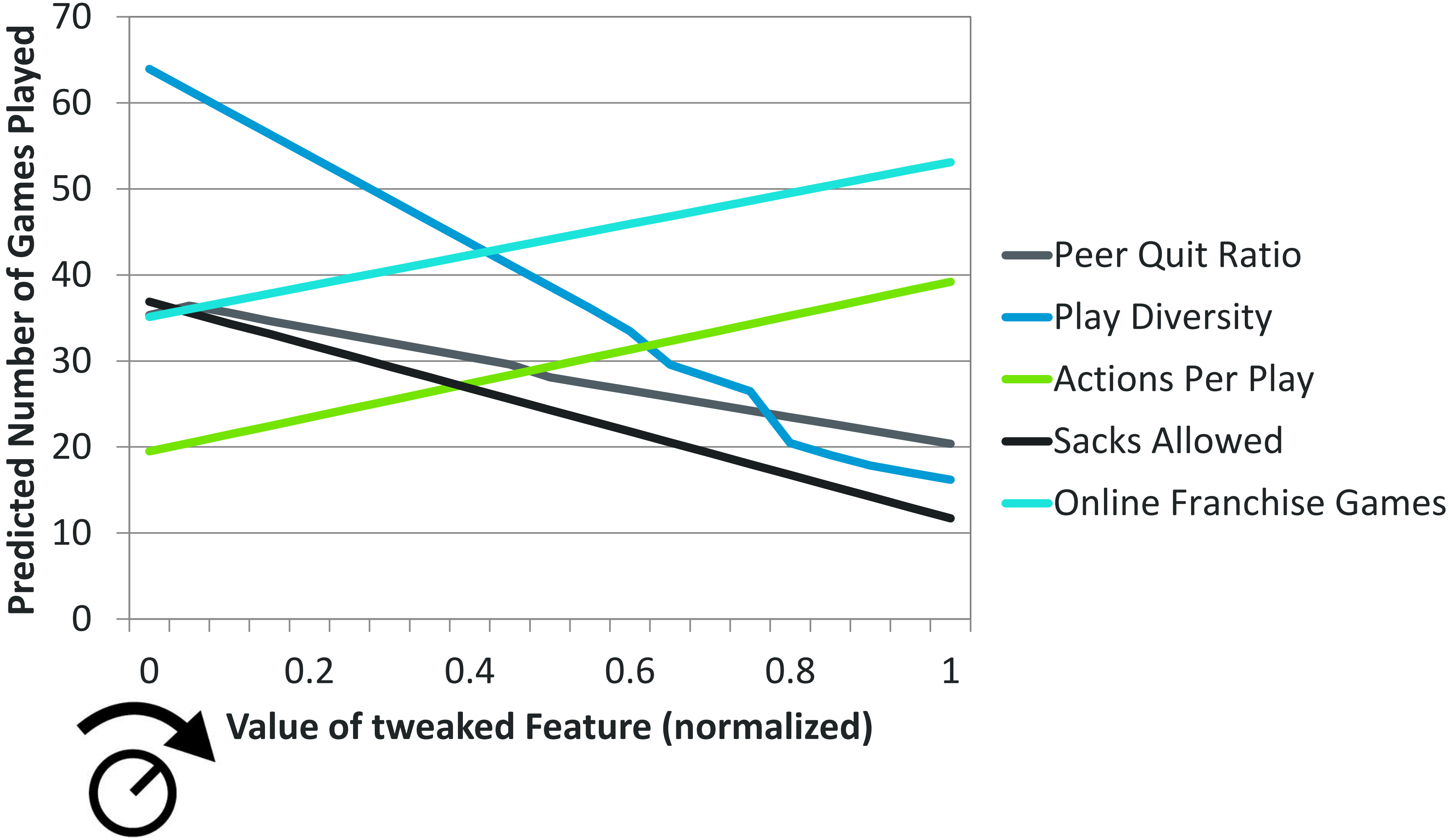
- Each player's behavior is encoded as the following features (46 total):
  - **Game modes**
    - Usage
    - Win rates
  - **Performance metrics**
    - Turnovers
    - Gain
  - **End conditions**
    - Completions
    - Peer quits
  - **Feature usage**
    - Gameflow
    - Scouting
    - Audibles
    - Special moves
  - **Play Preference**
    - Running
    - Play Diversity

Predicting the Number of Games Played



Feature Impact on Number of Games Played

- How does tweaking a single feature impact retention?



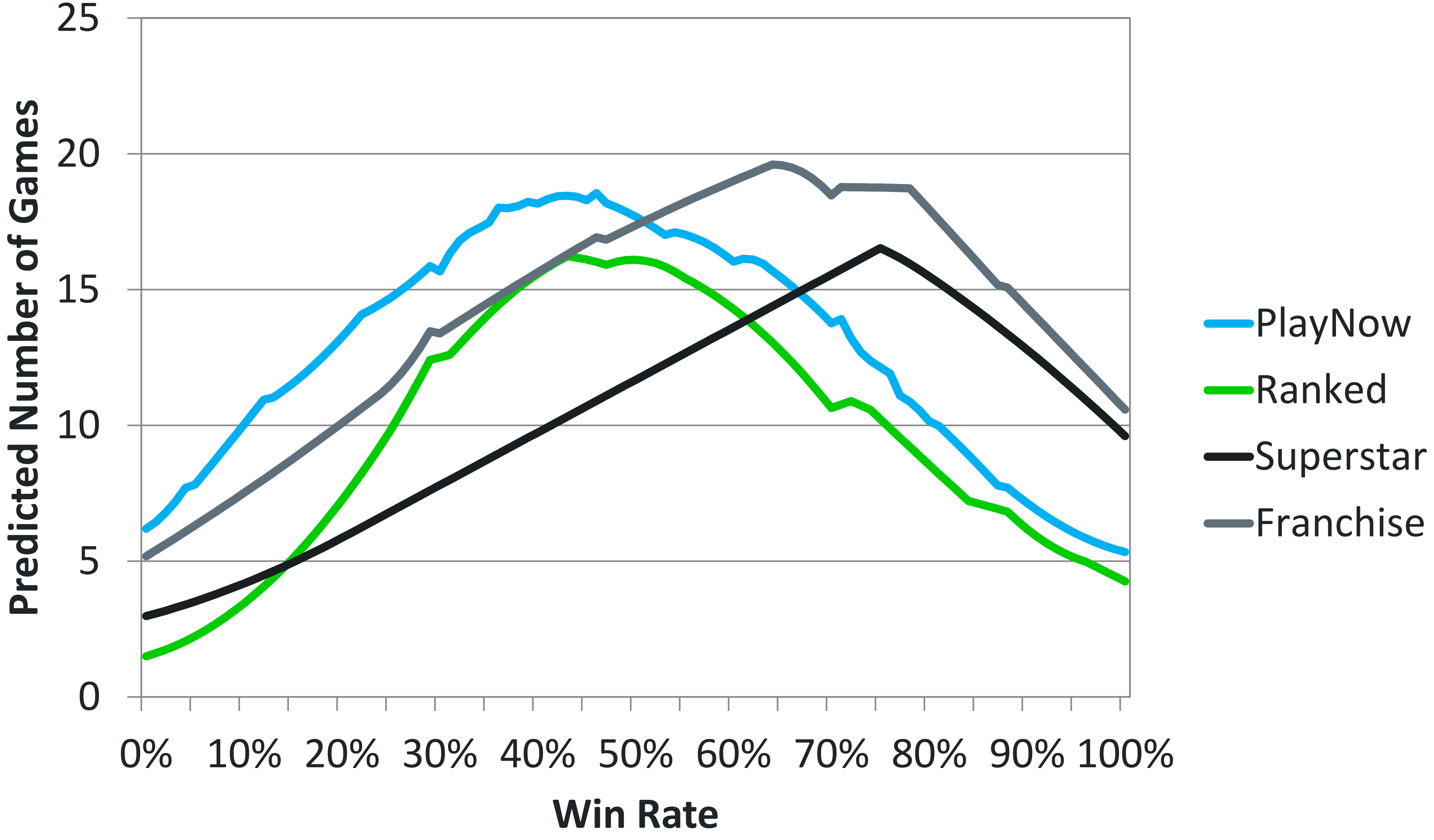
Most Influential Features

- The following features were identified as the **most influential** in predicting **player retention**

Correlation Strength ↑

Feature	Impact
Play Diversity	Negative
Online Franchise Wins	Positive
Running Plays	Positive
Sacks Made	Positive
Actions per Play	Positive
Interceptions Caught	Positive
Sacks Allowed	Negative
Peer Quit Ratio	Negative

Predicted Number of Games for Different Win Rates



Madden Project Findings

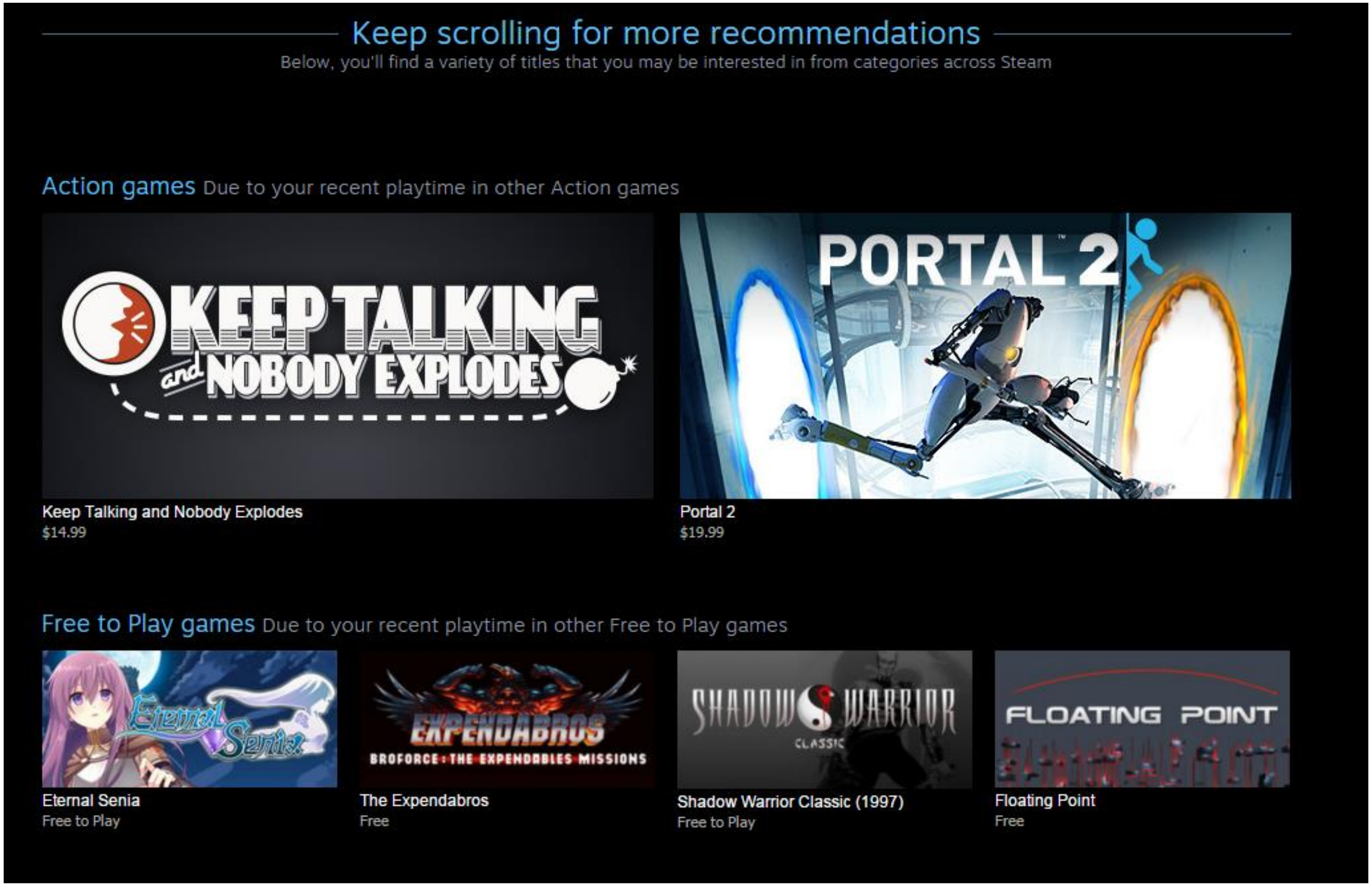
- **Simplify playbooks**
  - Players presented with a large variety of plays have lower retention and less success
- **Clearly present the controls**
  - Knowledge of controls had a larger impact than winning on player retention
- **Provide the correct challenge**
  - Multiplayer matches should be as even as possible, while single player should greatly favor the player

Recommendation Algorithms

- **Content-Based Filtering**
- **Collaborative Filtering**
  - Item-to-Item
  - User-to-User
- **Model Based**
  - Bayesian Inference

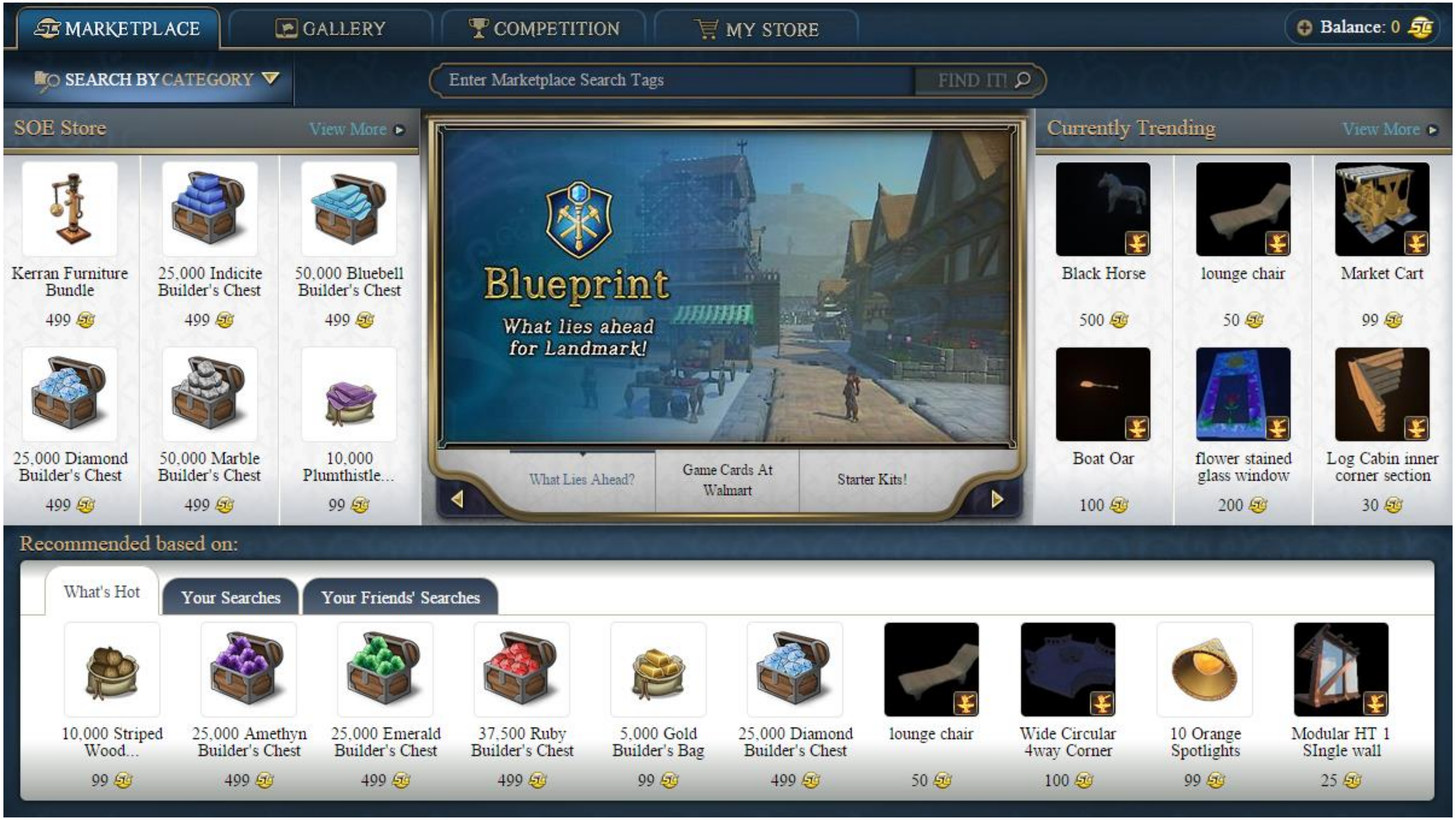
Content-Based Filtering

- Steam Storefront



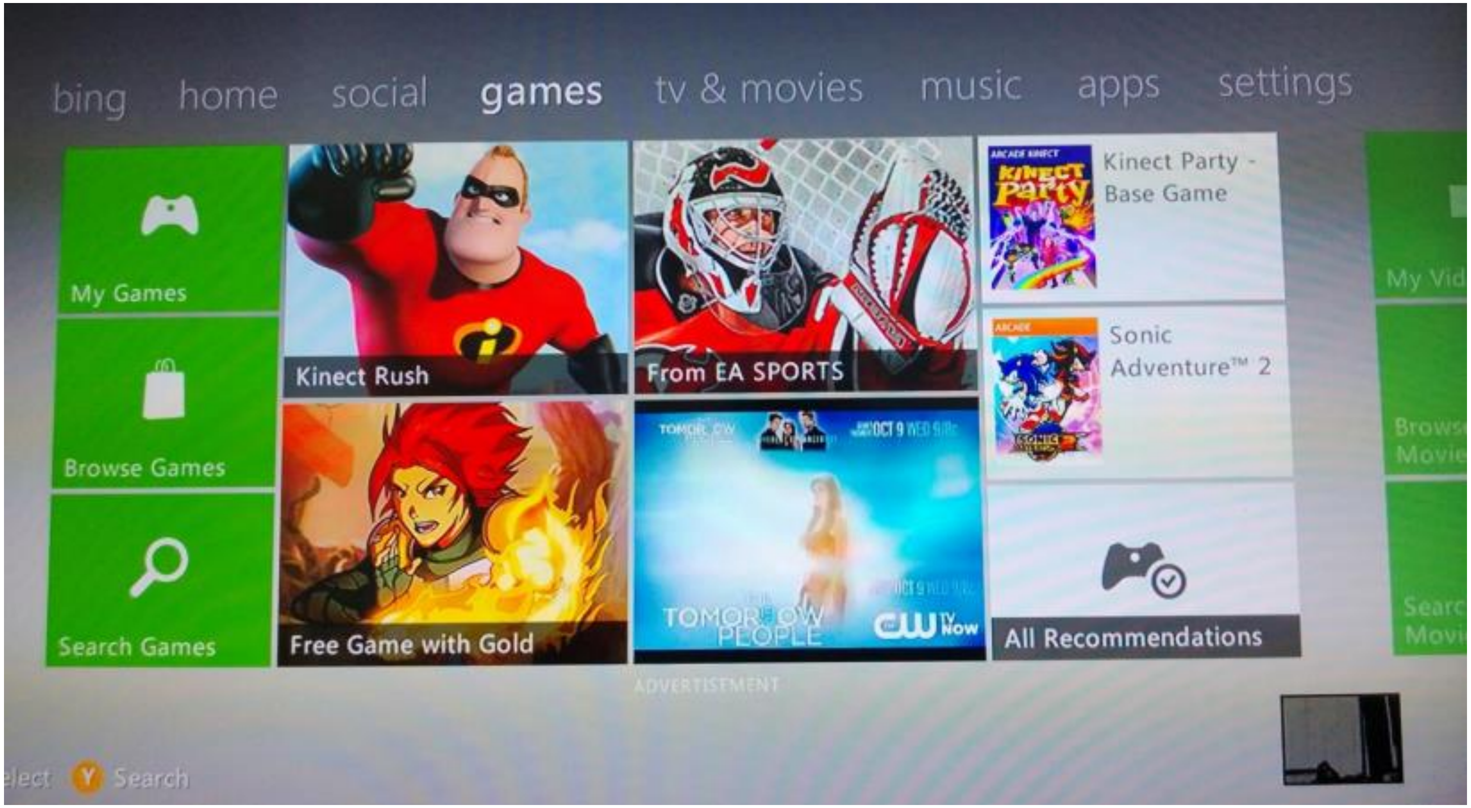
Collaborative Filtering

- EverQuest Landmark



Model Based

- Xbox Recommendation System



[Koenigstein et al., RecSys 2012]

Choosing an Algorithm

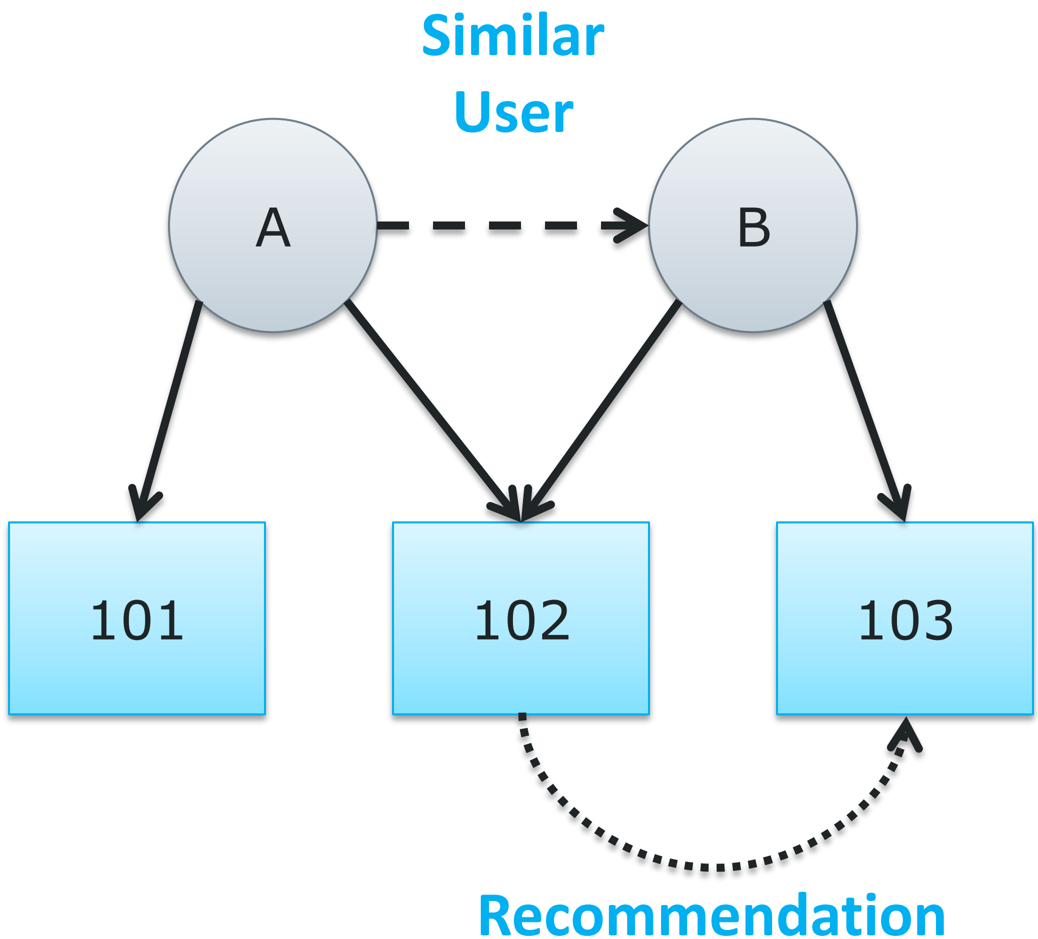
- **How big** is the item catalog? Is it **curated**?
- What is the target **number of users**?
- What **player context** will be used to provide item recommendations?

Collaborative Filtering (User-Based)

- Rates items for a player based on the player's **similarity** to **other players**
- Does not require **meta-data** to be maintained
- Can use **explicit** and **implicit** data collection
- Challenges include **scalability** and **cold starts**

User-Based Collaborative Filtering

- Users
- Items



Algorithm Overview

Computing a recommendation for a user, U:

For every other user, **V**

    Compute the similarity, **S**, between **U** and **V**

    For every item, **I**, rated by **V**

        Add **V**'s rating for **I**, weighted by **S** to a running average of **I**

Return the top rated items

Prototyping a Recommender

- **Apache Mahout**
  - Free & scalable Java machine learning library

- **Functionality**
  - User-based and item-based collaborative filtering
  - Single machine and cluster implementations
  - Built-in evaluation methods

Getting Started with Mahout

1. Choose what to recommend:
   ratings or **rankings**
2. Select a recommendation **algorithm**
3. Select a **similarity measure**
4. **Encode** your data into Mahout's format
5. **Evaluate** the results
6. Encode additional features and **iterate**

Generating Recommendations

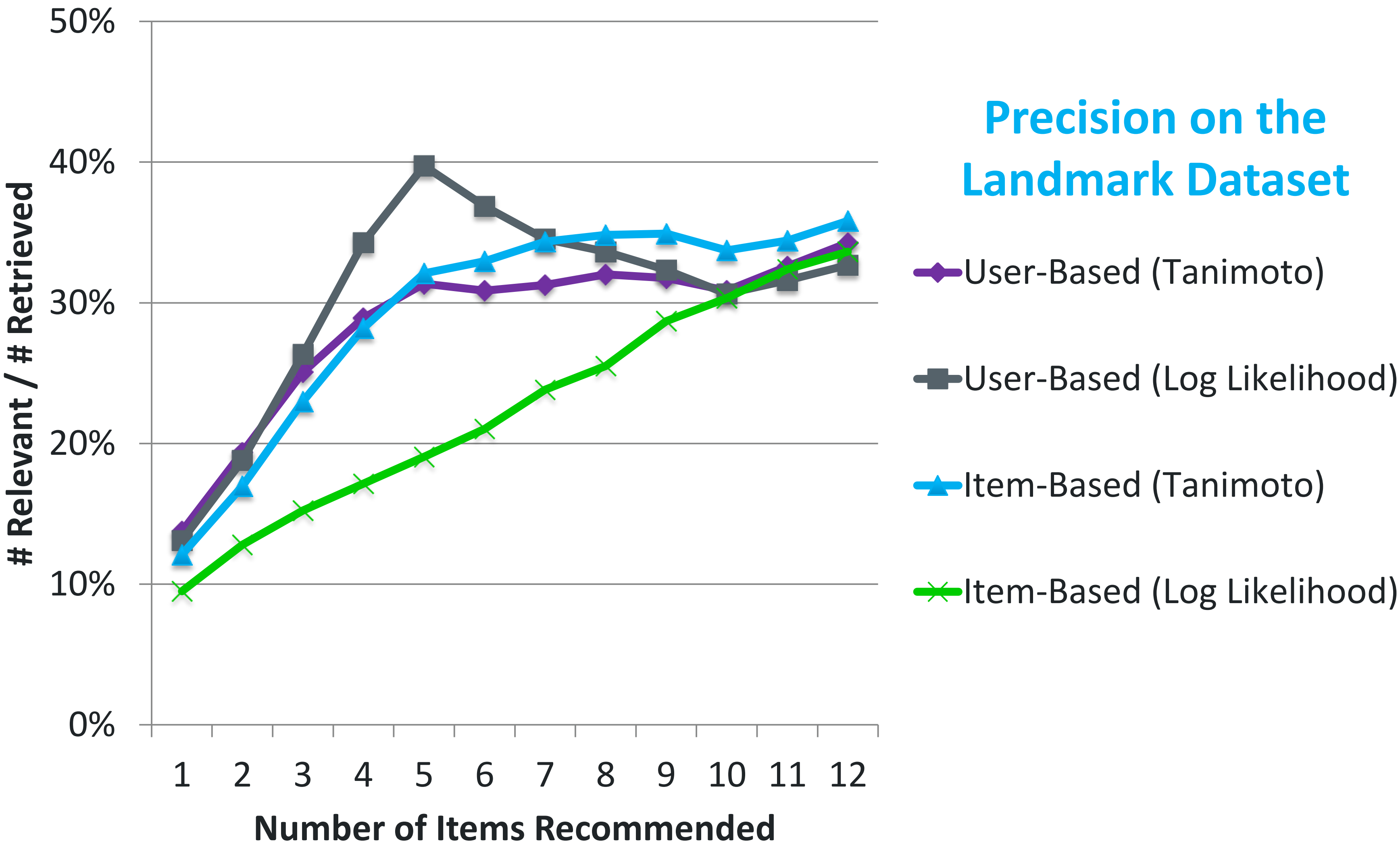
Building the Recommender

```
model = new DataModel(new File("SalesData.csv"));
similarity = new TanimotoSimilarity(model);
recommender  = new UserBasedRecommender(
                                        model, similarity);
```

Generating a List

```
recommendations = recommender.recommend(1, 6);
```

Evaluating Recommendations



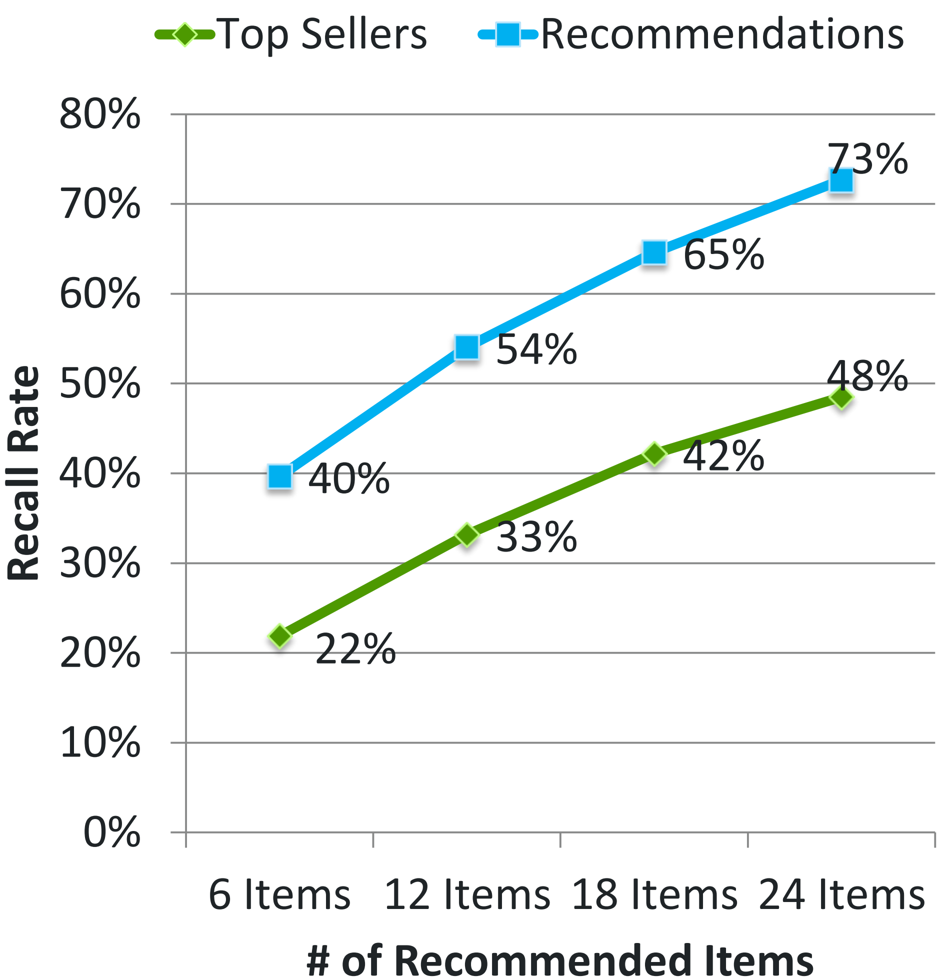
Holdout Experiment

- An experiment that **excludes** a **single item** from a player's list of purchases
- **Goals**
  - Generate the smallest list that includes the item
  - Enable offline evaluation of different algorithms
  - Compare recommendations with rule-based approaches

Landmark's Holdout Results

Recommendations significantly outperform a **top sellers** list

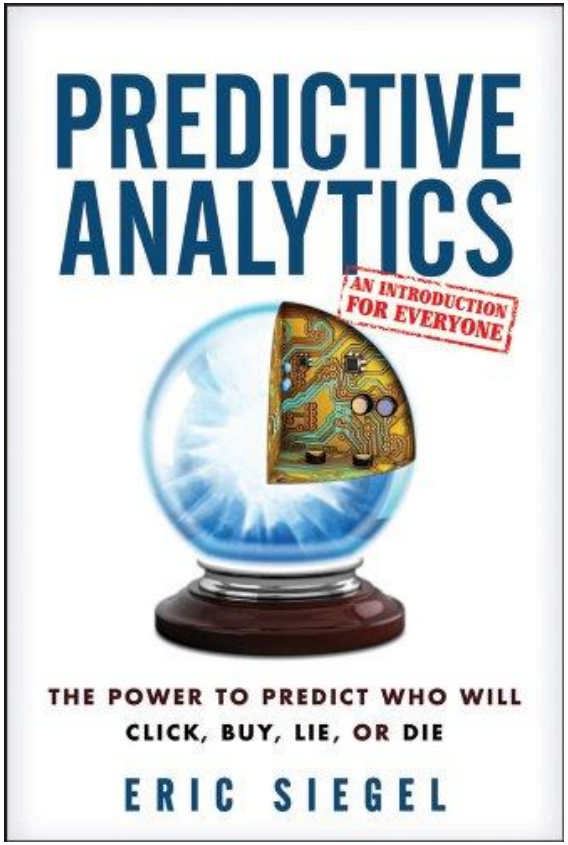
80% increase in the holdout **Recall Rate** at 6 items

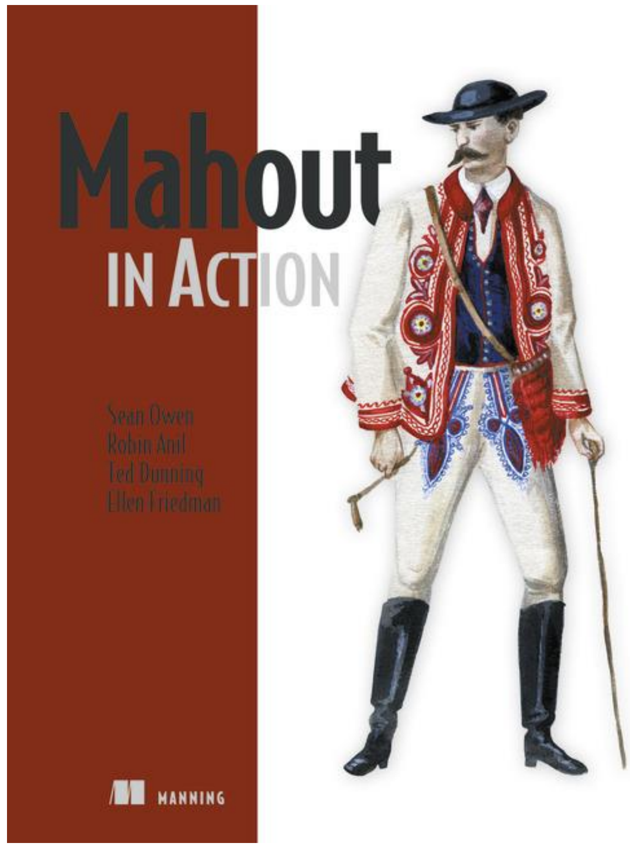


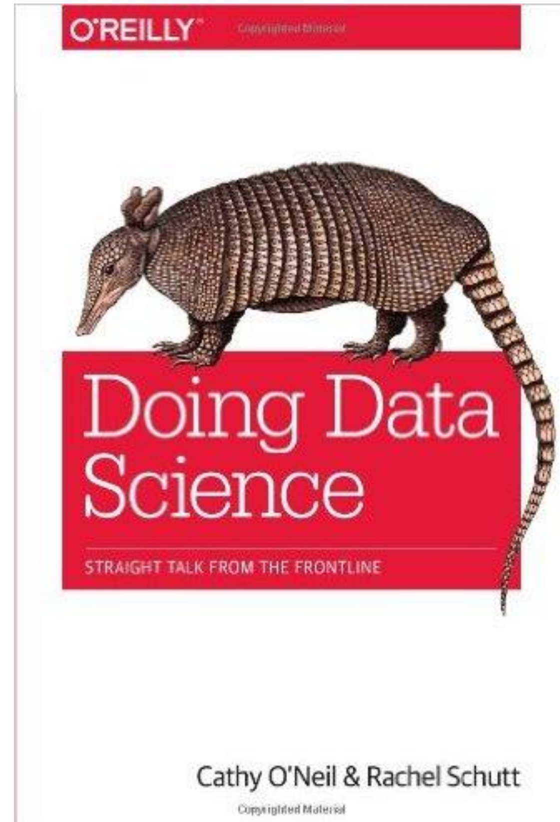
Deployment in Landmark

- **In-house implementation**
- **Current Deployment**
  - Recommendations are generated on the fly and cached
- **Planned Expansion**
  - An offline process builds a user-similarity matrix
  - An online process generates item recommendations in near real-time

Recommended Reading







Data Science at Electronic Arts

- Team Structure
- Technology Stack
- Project Lifecycles
- Career Path

My Projects at EA

- R Server & R Packages
- Origin & EA Access Analytics
- Analytics Best Practices
- Technology Innovation

Questions?

- **Ben G. Weber**
  - Senior Data Scientist, Electronic Arts
  - beweber@ea.com, @bgweber
  - http://tinyurl.com/WeberGameML
  - http://careers.ea.com