Killer Apps:
Data Mining Demystified

Mark Hornick

JSR 73/JSR 247 Specification Lead
Senior Manager
Oracle Corporation
www.oracle.com

TS-1262

Goal

Demystify how **data mining** technology can be used to create your own intelligent “killer app” through the

Java™ Specification Request (JSR) 73 Data Mining API

Agenda

Exploring Data Mining

Killer Apps for Data Mining

Building a Data Mining Application

JSR 73 and JSR 247

Agenda

Exploring Data Mining

Killer Apps for Data Mining

Building a Data Mining Application

JSR 73 and JSR 247

What Is Data Mining?

- Extracting actionable knowledge and insight from data
- Also known as…
  - Advanced Analytics
  - Predictive Analytics
  - Machine Learning
- Foundations in statistics, mathematics, machine learning, and other sciences
- A key component to “Business Intelligence”

Business Intelligence

Query and Reporting	OLAP	Data Mining
Extraction of detailed and roll up data	Summaries, trends and forecasts	Knowledge discovery of hidden patterns
“Information”	“Analysis”	“Knowledge and Insight”
Which customers rented sci-fi movies last year?	**What is the** average movie **rating** of sci-fi renters, by genre, by region?	**Who is likely to** rent ***this*** new sci-fi movie next month and why?

What Is a Model?

- A compact representation of knowledge or patterns present in data
- Produced from a variety of techniques…

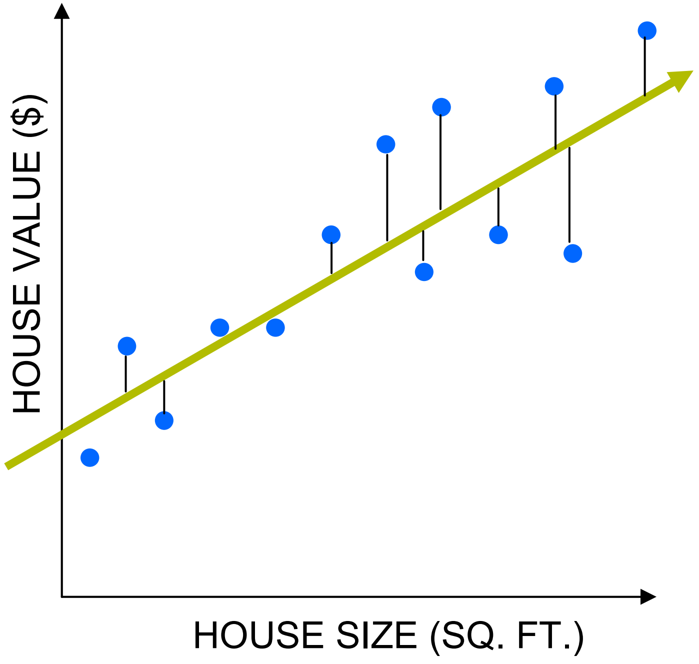
  Regression, **Classification**, Clustering, **Association**

  Attribute Importance, Anomaly Detection, Time Series,

  Feature Extraction, Text Mining, Sequence Mining
- A model can predict values in a generalized way

Regression

Predict a continuous numerical value



For a simple dataset with two attributes, a line can be used to approximate the values

$$y = mx + b$$

A simple *model* can be expressed in terms of values (m, b)

Models aren’t perfect… predictions have an error component

Metrics like Root Mean Square Error (RMSE) are useful for assessing and comparing models

Why Data Mining Models?

- Consider large datasets
  - 100s or 1000s attributes
  - 1000s to millions of records
  - Some are strings, others are numbers
  - Some have ordered values, others have unordered values
- It is intractable for a person to identify patterns or extract knowledge from such a large dataset
- But a computer and the right algorithm can do so very efficiently

Agenda

Exploring Data Mining

Killer Apps for Data Mining

Building a Data Mining Application

JSR 73 and JSR 247

What Is a “killer app?”

- “…a computer program that is so useful or desirable that it proves the value of some underlying technology, such as a gaming console, operating system, or piece of computer hardware.”

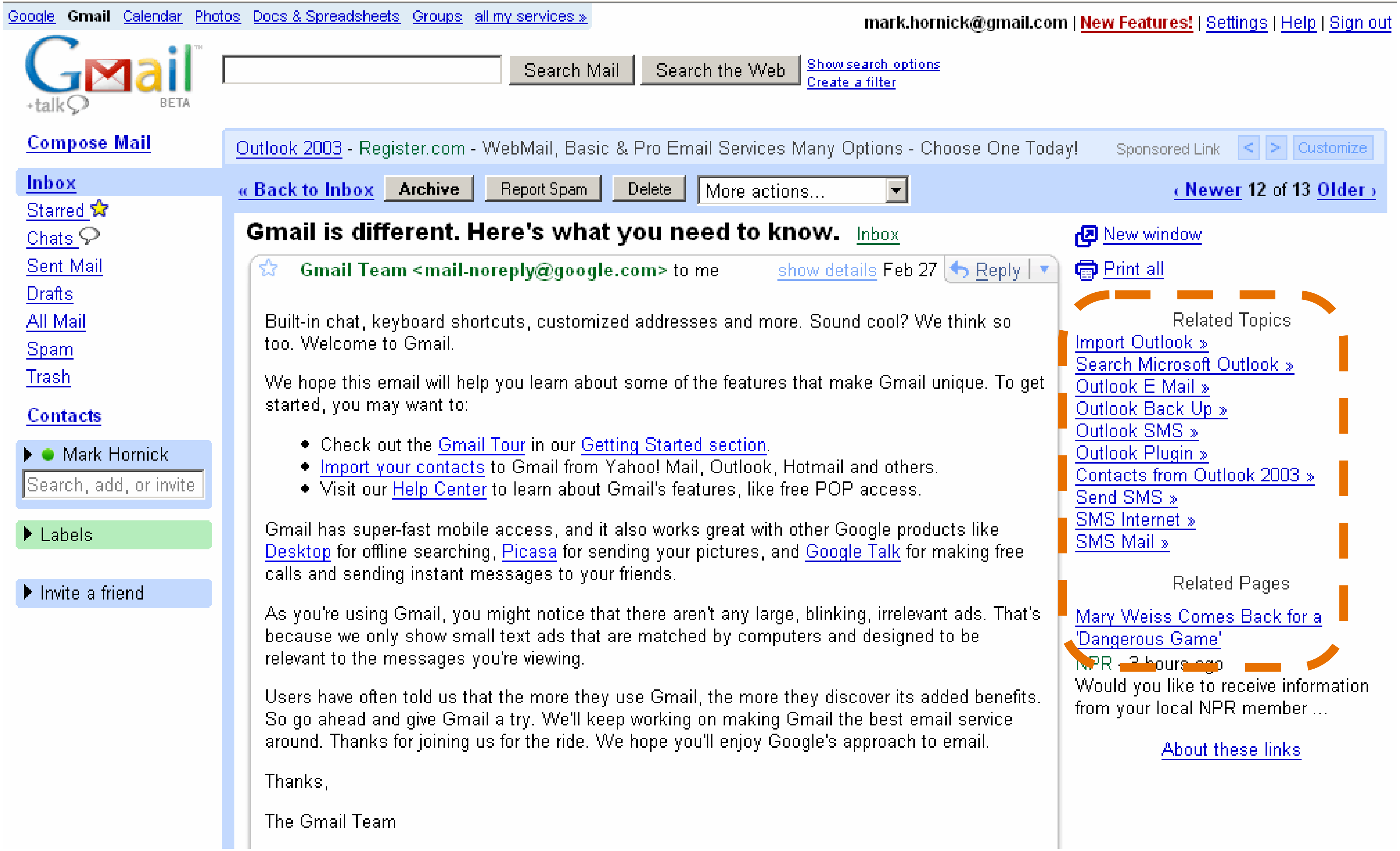
…or, advanced analytics software like data mining!

Source: http://en.wikipedia.org/wiki/Killer_app

Disclaimer

The approaches proposed here reflect how these features could be realized using data mining

Actual realizations may involve non-data mining techniques or a combination of techniques



Gmail is different. Here's what you need to know. Inbox

Gmail Team <mail-noreply@google.com> to me — show details Feb 27 — Reply

Built-in chat, keyboard shortcuts, customized addresses and more. Sound cool? We think so too. Welcome to Gmail.

We hope this email will help you learn about some of the features that make Gmail unique. To get started, you may want to:

- Check out the Gmail Tour in our Getting Started section.
- Import your contacts to Gmail from Yahoo! Mail, Outlook, Hotmail and others.
- Visit our Help Center to learn about Gmail's features, like free POP access.

Gmail has super-fast mobile access, and it also works great with other Google products like Desktop for offline searching, Picasa for sending your pictures, and Google Talk for making free calls and sending instant messages to your friends.

As you're using Gmail, you might notice that there aren't any large, blinking, irrelevant ads. That's because we only show small text ads that are matched by computers and designed to be relevant to the messages you're viewing.

Users have often told us that the more they use Gmail, the more they discover its added benefits. So go ahead and give Gmail a try. We'll keep working on making Gmail the best email service around. Thanks for joining us for the ride. We hope you'll enjoy Google's approach to email.

Thanks,

The Gmail Team

Related Topics

- Import Outlook »
- Search Microsoft Outlook »
- Outlook E Mail »
- Outlook Back Up »
- Outlook SMS »
- Outlook Plugin »
- Contacts from Outlook 2003 »
- Send SMS »
- SMS Internet »
- SMS Mail »

Related Pages

Mary Weiss Comes Back for a 'Dangerous Game'
NPR - 3 hours ago
Would you like to receive information from your local NPR member ...

About these links

How Can DM Be Used to Do This?

- Use text analysis to extract terms and themes from email
- Record user responses to ads against extracted terms and themes
- Predict which ads are most likely interesting to email recipient, based on…
    - Email content
    - User profile and previous actions

Classification Technique

- Build classification model using historical data on persons, email content, and known ad clicks
- Predict if **this** person will “click” or “not click” **this** ad given email content and profile
- Rank ads according to probability of being clicked
- Select the top N to display to user

Classification

Predict category and probability for each case

Historical data for Ad-23 with known outcomes

Cases

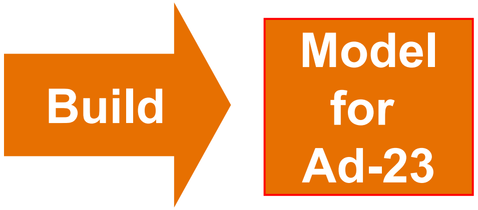
User	Income	Age	. . .		Clicked on ad?
236	30,000	30			Yes
681	55,000	67			Yes
372	25,000	23			No
493	50,000	44			No

$$X_1 \qquad \underbrace{X_2 \qquad ...... \qquad X_m} \qquad Y$$

Key Attribute

Predictor Attributes

Target Attribute



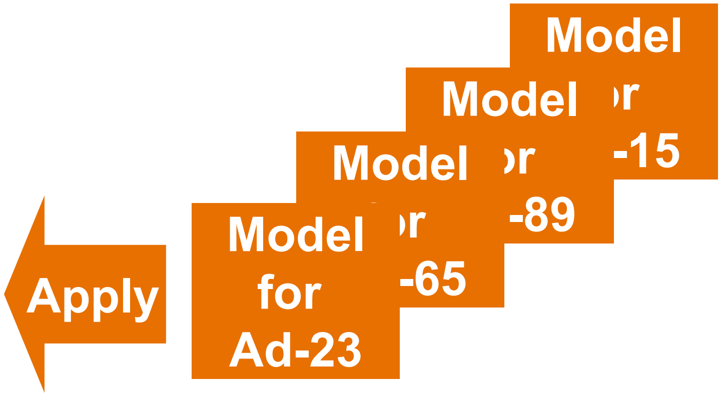
Build a model to create a Functional Relationship:

$$Y = F(X_2, X_3, \ldots, X_m)$$

Classification

Apply models to data

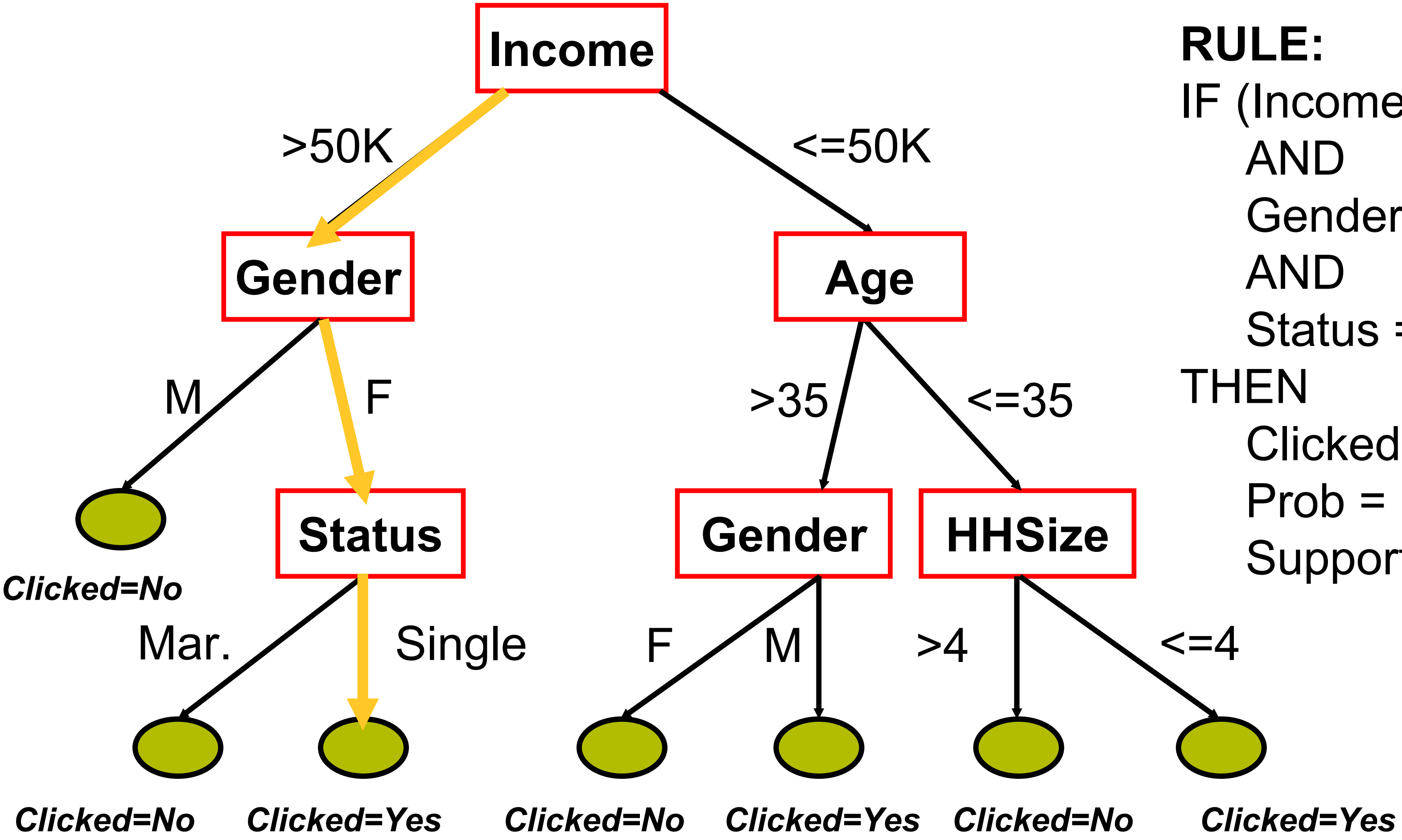
Name	Income	Age	. . .	Ad#	Prob of Clicked="Yes"
572	40,500	52		23	.86
				65	.23
				89	.95
				15	.34



- Apply each model
- Order ads by probability
- Select top N

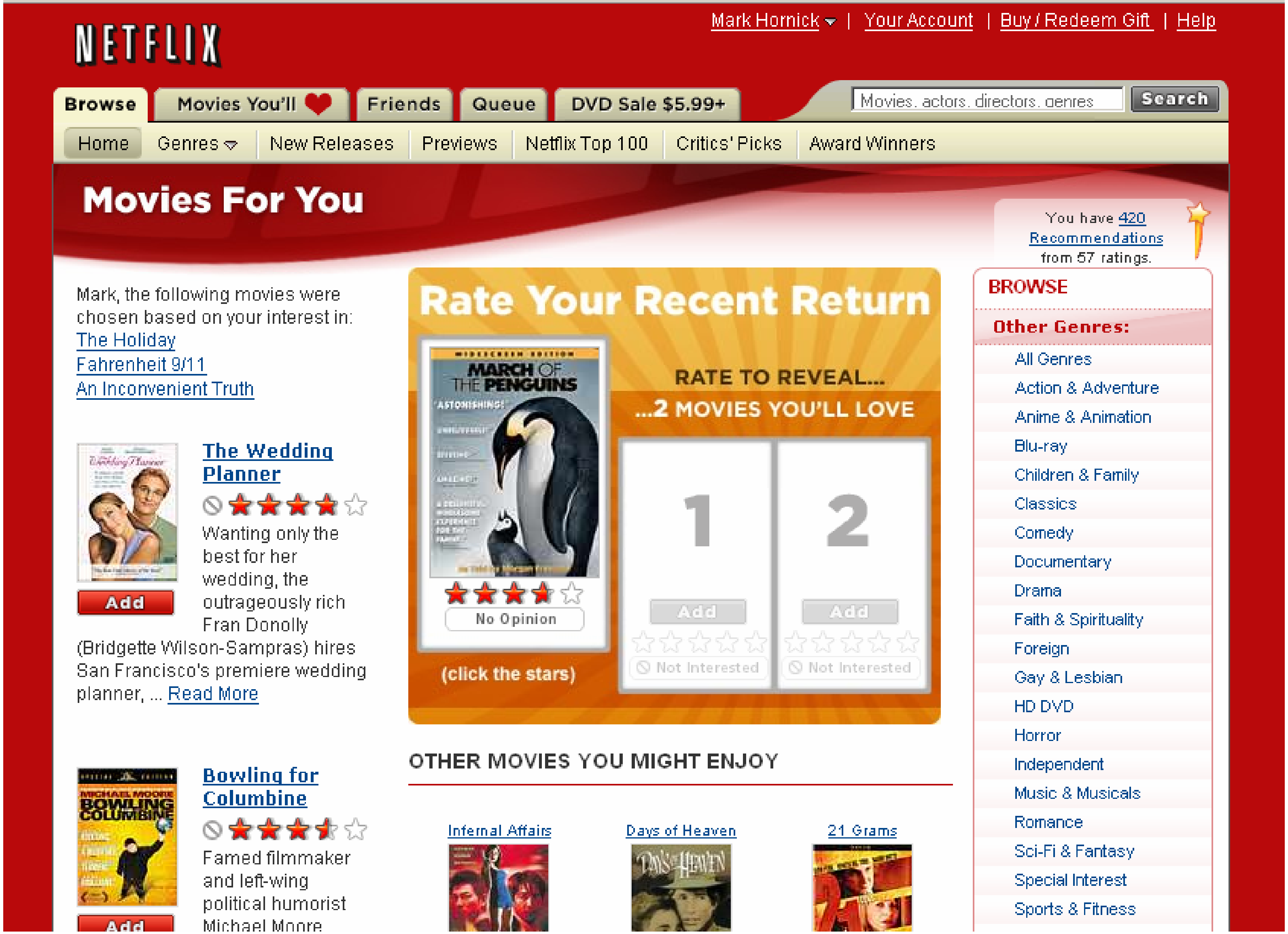
Algorithm: Decision Tree

Produces a “tree model”



RULE:

IF (Income >50K
AND
Gender = F
AND
Status = Single)
THEN
Clicked = Yes
Prob = .77
Support = .15



NETFLIX

Mark Hornick | Your Account | Buy / Redeem Gift | Help

Movies For You

You have 420 Recommendations from 57 ratings.

Mark, the following movies were chosen based on your interest in:
The Holiday
Fahrenheit 9/11
An Inconvenient Truth

The Wedding Planner

Wanting only the best for her wedding, the outrageously rich Fran Donolly (Bridgette Wilson-Sampras) hires San Francisco's premiere wedding planner, ... Read More

Bowling for Columbine

Famed filmmaker and left-wing political humorist Michael Moore

Rate Your Recent Return

RATE TO REVEAL...
...2 MOVIES YOU'LL LOVE

OTHER MOVIES YOU MIGHT ENJOY

Infernal Affairs
Days of Heaven
21 Grams

BROWSE

Other Genres:

- All Genres
- Action & Adventure
- Anime & Animation
- Blu-ray
- Children & Family
- Classics
- Comedy
- Documentary
- Drama
- Faith & Spirituality
- Foreign
- Gay & Lesbian
- HD DVD
- Horror
- Independent
- Music & Musicals
- Romance
- Sci-Fi & Fantasy
- Special Interest
- Sports & Fitness

How Can DM Be Used to Do This?

- Track user actions and ratings
  - Data: customer, movie, rating
- Build classification models to predict whether user will like movie and with what probability
  - One model per movie
  - 10,000 movies → 10,000 models
- Build an association model to get rules

Association (Market Basket Analysis)

Transactional Data and Rule Example

Input Data:

User ID	Movies Viewed
1	{Movie1, Movie2, Movie3}
2	{Movie1, Movie4}
3	{Movie1, Movie3}
4	{Movie2, Movie5, Movie6}
…	…
N	{Movie3, Movie4, Movie6}

Movie1 and Movie2 → Movie3

with support of .12 and confidence .78

Association Rules

Support and Confidence

User ID	Movies Viewed
1	{1, 2, 3}
2	{1, 4}
3	{1, 3}
4	{2, 5, 6}

Support (A → B)
= P(AB)
= count (A & B) / totalCount

Confidence (A → B)
= P(AB)/P(A)
= count (A & B) / count (A)

1 → 3 :

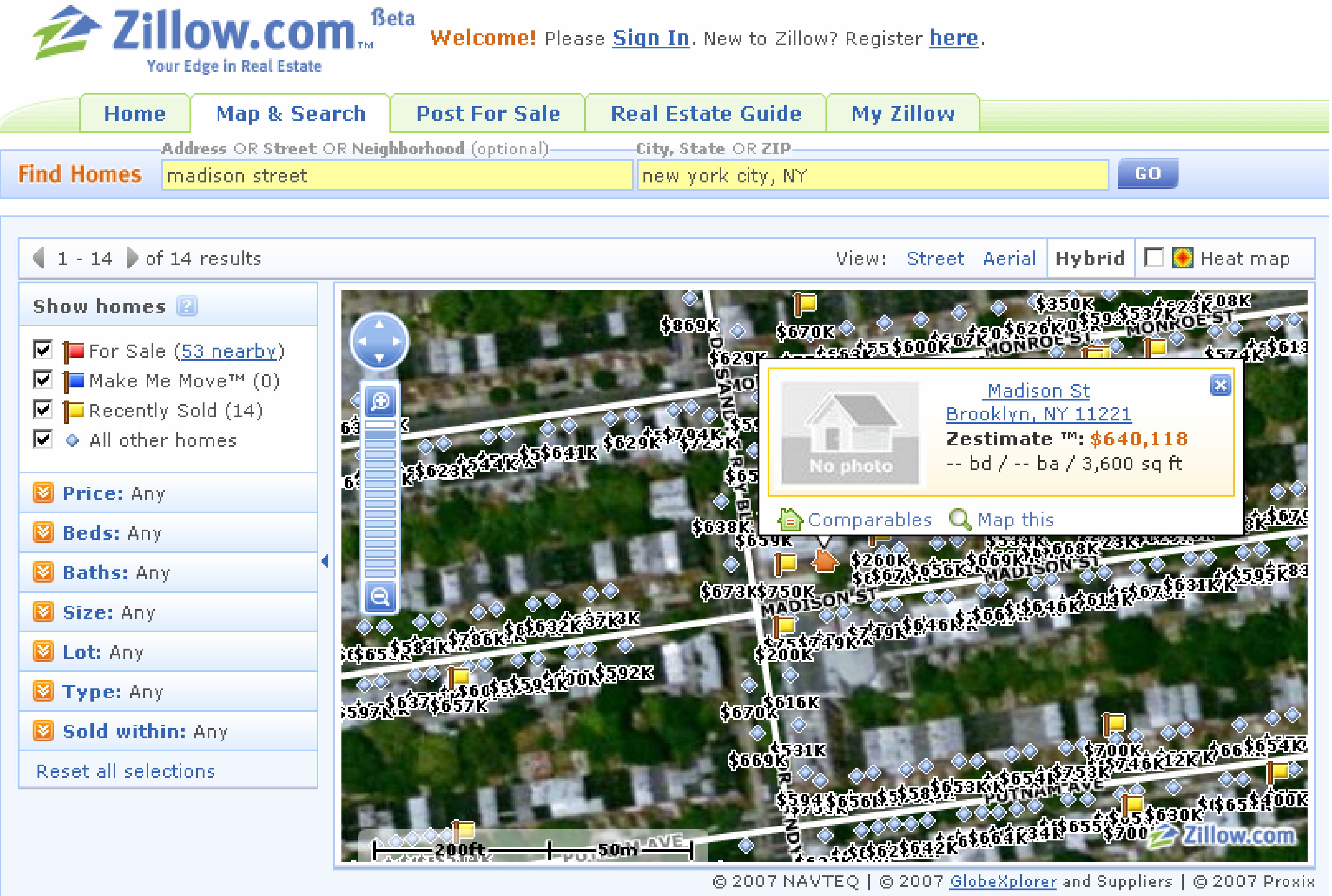
Support = 2/4 = 50%

Confidence = 2/3 = 66%

3 → 1 :

Support = 2/4 = 50%

Confidence = 2/2 = 100%



Zillow.com™ Beta
Your Edge in Real Estate

Welcome! Please Sign In. New to Zillow? Register here.

Home | Map & Search | Post For Sale | Real Estate Guide | My Zillow

Find Homes
Address OR Street OR Neighborhood (optional): madison street
City, State OR ZIP: new york city, NY
GO

1 - 14 of 14 results   View: Street Aerial Hybrid □ Heat map

Show homes
☑ For Sale (53 nearby)
☑ Make Me Move™ (0)
☑ Recently Sold (14)
☑ All other homes

Price: Any
Beds: Any
Baths: Any
Size: Any
Lot: Any
Type: Any
Sold within: Any
Reset all selections

Madison St
Brooklyn, NY 11221
Zestimate ™: $640,118
-- bd / -- ba / 3,600 sq ft
No photo
Comparables   Map this

© 2007 NAVTEQ | © 2007 GlobeXplorer and Suppliers | © 2007 Proxix

How Can DM Be Used to Do This?

- Collect relevant data
  - Multiple Listing Service (MLS) data on properties
  - Actual sale prices, days on market,…
- Build a regression model to predict property values based on property attributes and known sale price
- Periodically rebuild the model with additional sales data
- Score homes in batch so predictions are ready
- Real-time score to reflect online changes

Agenda

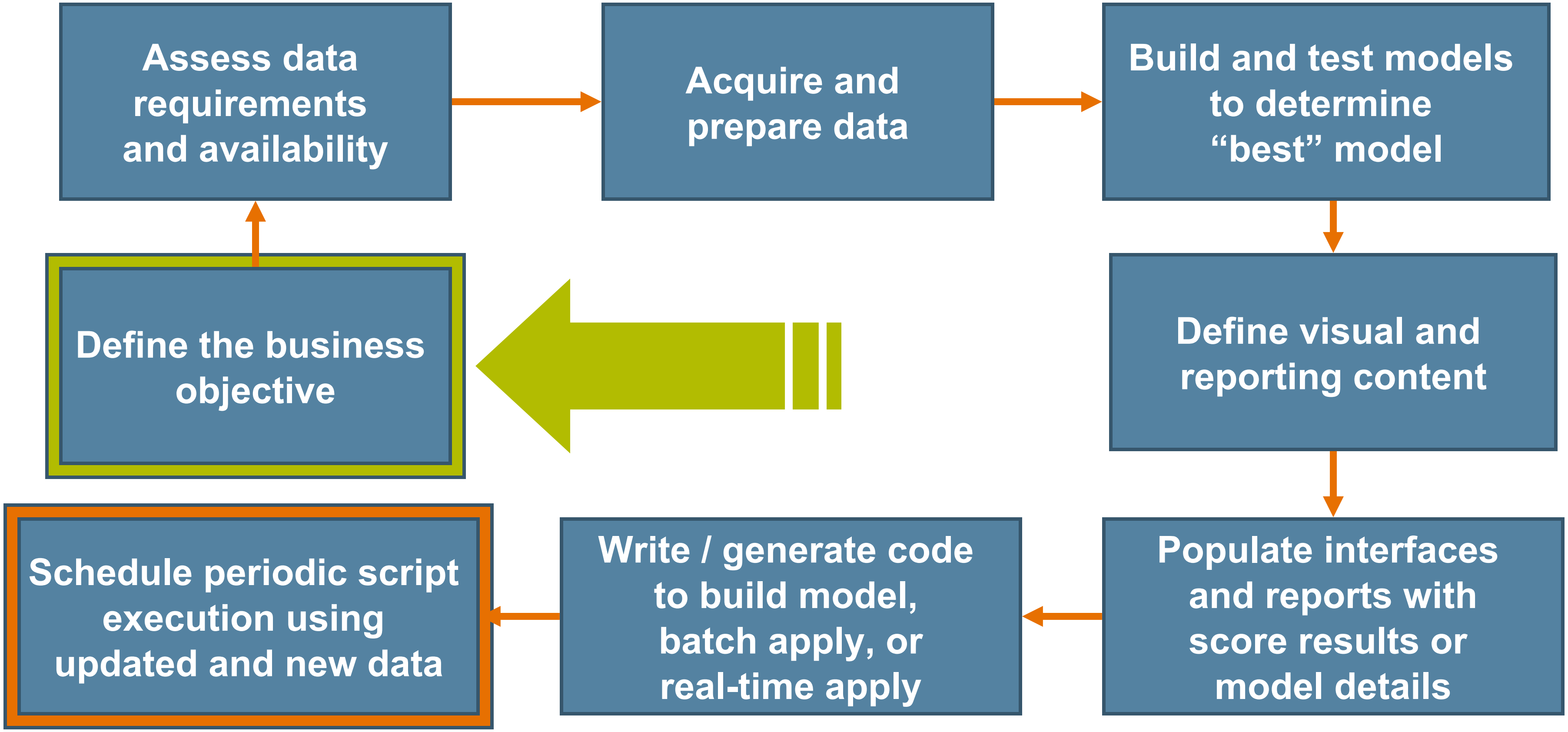
Exploring Data Mining

Killer Apps for Data Mining

Building a Data Mining Application

JSR 73 and JSR 247

Building a Data Mining Application



Define the business objective

Assess data requirements and availability

Acquire and prepare data

Build and test models to determine “best” model

Define visual and reporting content

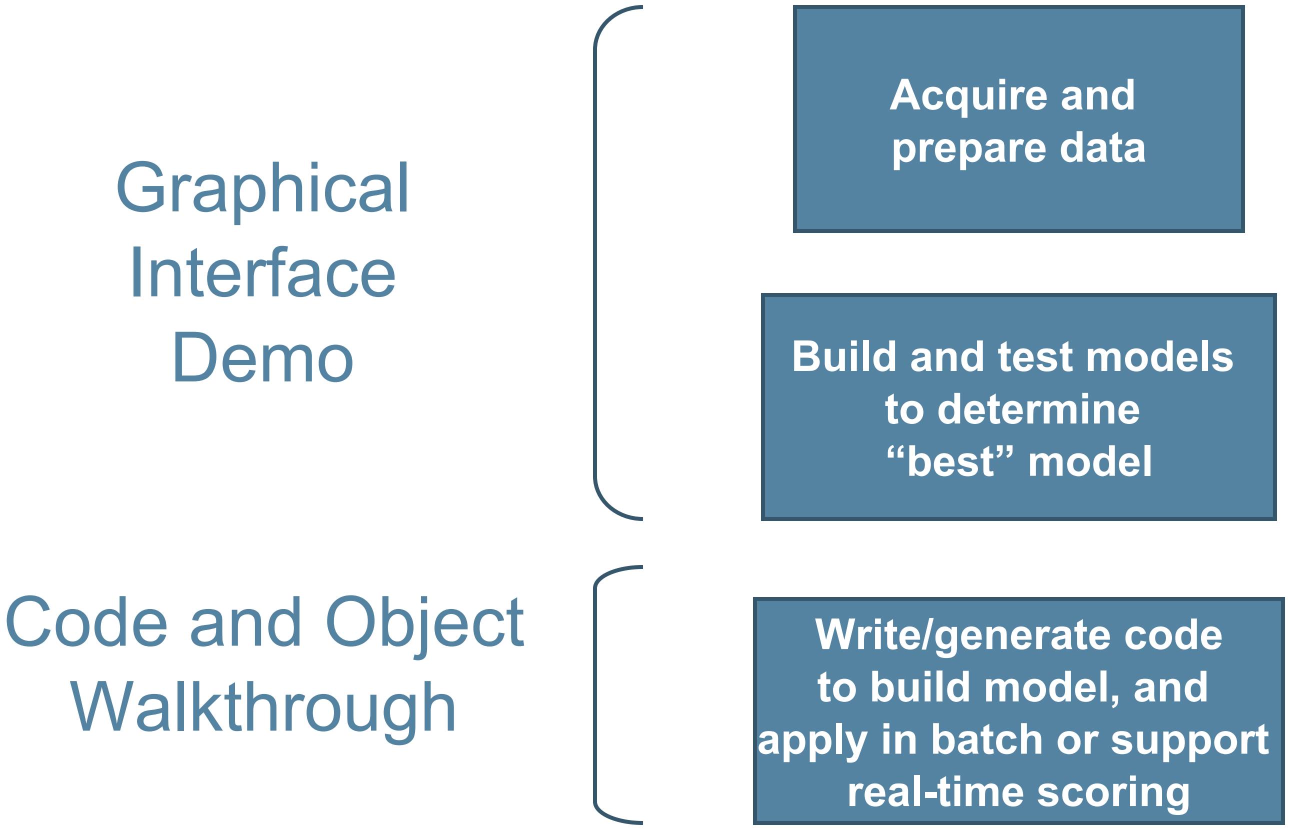
Populate interfaces and reports with score results or model details

Write / generate code to build model, batch apply, or real-time apply

Schedule periodic script execution using updated and new data

Building a Data Mining Application

We’ll focus on these steps…



Graphical Interface Demo

Acquire and prepare data

Build and test models to determine “best” model

Code and Object Walkthrough

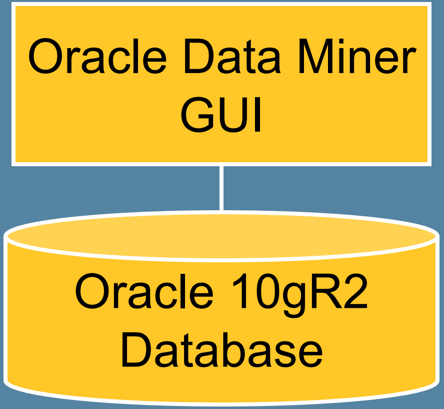
Write/generate code to build model, and apply in batch or support real-time scoring

Data Preparation

May involve…

- Joining multiple tables, from multiple sources
- Transforming data
  - Data cleansing
  - Business transformations
    e.g., computed attribute `income/age^2`,
    bin `age` into 5 bins, recode `9999` to `NULL`,
    aggregate `items` to `item_count`
  - Algorithm-required transformations
    e.g., binning, normalization, outlier and missing value treatment

Transformations omitted in JSR 73, defined in JSR 247



Oracle Data Miner GUI

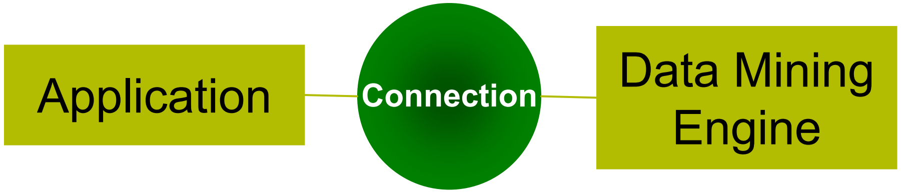
Oracle 10gR2 Database

DEMO

Build, test, and apply a data mining model to predict house values using a graphical interface that uses JSR 73

JDM Connection Object

- Used to interact with Data Mining Engine (DME)
- Obtained via vendor class or Java Naming and Directory Interface™ (JNDI) API
- Connection provides methods to
  - Support object lifecycle management (save, remove, etc.)
  - Selectively retrieve objects
  - Execute tasks and obtain task execution status



Getting a Connection Factory Using JNDI API

```
// Using JNDI to get the Connection Factory
Hashtable env = new Hashtable();
env.put( Context.INITIAL_CONTEXT_FACTORY,
         “com.myCompany.javax.datamining.
                   resource.initialContextFactory-Impl" );
env.put( Context.PROVIDER_URL,"http://myHost:myPort/myService");
env.put( Context.SECURITY_PRINCIPAL, "user" );
env.put( Context.SECURITY_CREDENTIALS, "password" );

InitialContext jndiContext =
      new javax.naming.InitialContext( env );

// Perform JNDI lookup to obtain the connection factory
javax.datamining.resource.ConnectionFactory m_dmeConnFactory =
      (ConnectionFactory) jndiContext.lookup(
                                 "java:comp/env/jdm/MyServer");
```

Getting a Vendor-Specific Connection

Application
Connection
Data Mining Engine

```
// Login to the Data Mining Engine
m_dmeConnFactory = new OraConnectionFactory();
ConnectionSpec cs = m_dmeConnFactory.getConnectionSpec();

cs.setURI("jdbc:oracle:thin:@"+uri);
cs.setName(name);
cs.setPassword(password);
m_dmeConn = m_dmeConnFactory.getConnection(cs);
```

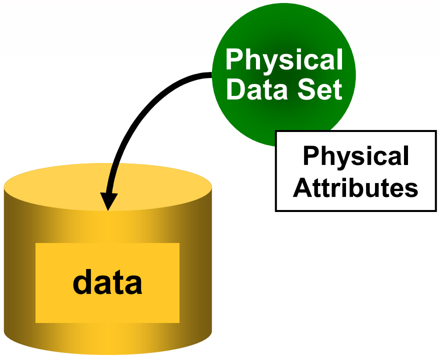
Obtain JDM Factories

```
// Obtain factories for needed objects
m_pdsFactory = (PhysicalDataSetFactory)
      m_dmeConn.getFactory(
                "javax.datamining.data.PhysicalDataSet");

m_pdrFactory = (PhysicalDataRecordFactory)
      m_dmeConn.getFactory(
                "javax.datamining.data.PhysicalDataRecord");

m_paFactory = (PhysicalAttributeFactory)
      m_dmeConn.getFactory(
                "javax.datamining.data.PhysicalAttribute");

// ...
```

JDM PhysicalDataSets

- Users reference data table or file through a “PhysicalDataSet” object
  - Maps columns to physical attribute objects
  - Describes how to interpret columns
  - Identifies role, e.g., data, transaction ID, case ID
- Specifies URI of data for DME to access
- Provides methods to extract table metadata
  - Populate PhysicalAttribute objects
  - Include attribute name, datatype, comments

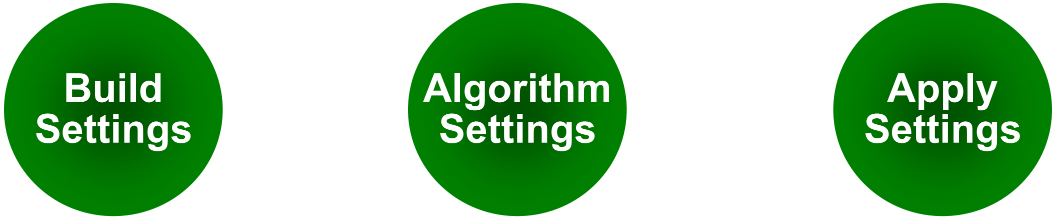
Create the Build Dataset



```
// Create the physical dataset object
PhysicalDataSet buildData =
    m_pdsFactory.create("BOSTON_HOUSING_BUILD_SVM",
                        NO_METADATA);

// Create a physical attribute to specify the ID
PhysicalAttribute pa =
    m_paFactory.create("ID",
                       AttributeDataType.integerType,
                       PhysicalAttributeRole.caseId );

buildData.addAttribute(pa);

// Save the object
m_dmeConn.saveObject("svmrBuildData_jdm",
                     buildData, REPLACE);
```

JDM Settings Objects

- Capture parameters that control mining operations
- BuildSettings objects
  - Specific to each mining function and algorithm
  - Optionally specifies algorithm settings
  - Optionally specifies logical interpretation of attributes
- ApplySettings objects
  - Used when scoring to determine result content

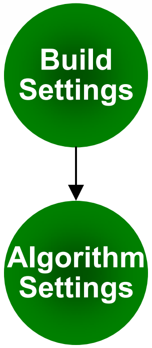


Build Settings

Algorithm Settings

Apply Settings

Create the Build Settings



```
// Create regression build settings
RegressionSettings buildSettings = m_regrFactory.create();

// Create SVM Regression algorithm settings
SVMRegressionSettings svmrAlg = m_svmrFactory.create();

svmrAlg.setKernelFunction(KernelFunction.kLinear);

// Assign the algorithm settings
buildSettings.setAlgorithmSettings(svmrAlg);

// Specify the target attribute - home value
buildSettings.setTargetAttributeName("MEDV");

// Save the object
m_dmeConn.saveObject("svmrBuildSettings_jdm",
                     buildSettings, REPLACE);
```

JDM Task Objects

- Container for specifying inputs to mining operations
- Supports synchronous and asynchronous execution
- Executing a task produces a handle for checking status of or terminating executing tasks

Build Task

Build Data
Build Settings
Model Name
➔ Model

Test Task

Test Data
Test Metrics
Model Name
➔Test Results

Apply Task

Apply Data
Apply Settings
Model Name
Apply Result URI
➔ Scored Data

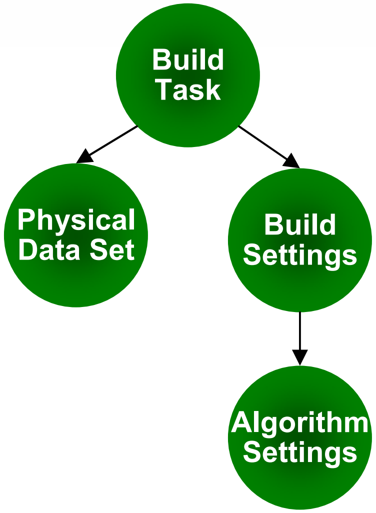
Import Task

URI of objects
➔imported set of objects

Export Task

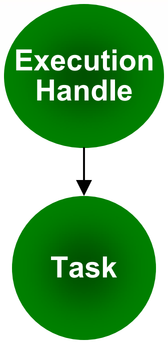
Object Set
Settings
URI
➔ Destination contains Exported objects

Create the Build Task



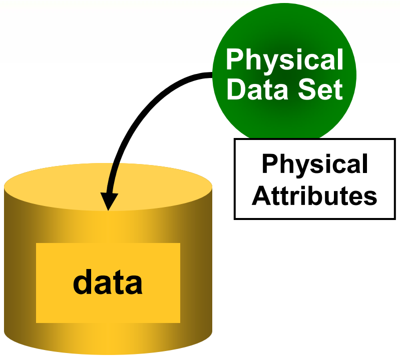
```
// Create the build task
BuildTask buildTask = m_buildFactory.create(
         "svmrBuildData_jdm",      //Build data
         "svmrBuildSettings_jdm", //Mining settings
         "svmrModel_jdm“);         //Output: Mining model

// Save the build task with a name
m_dmeConn.saveObject("svmrBuildTask_jdm”,
                     buildTask, REPLACE);
```

Execute the Build Task



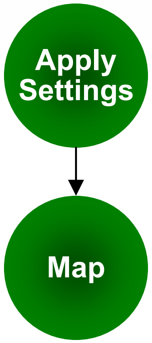
```
//Execute the task asynchronously
ExecutionHandle execHandle =
            m_dmeConn.execute("svmrBuildTask_jdm”);

//Wait for completion of the task
ExecutionStatus status =
            execHandle.waitForCompletion(Integer.MAX_VALUE);

//Check the status of the task after completion
boolean isTaskSuccess =
            status.getState().equals(ExecutionState.success);
```

Create the Apply Dataset



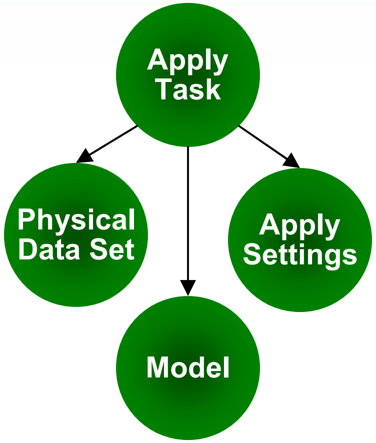
```
// Create reference object for the scoring dataset
PhysicalDataSet applyData =
      m_pdsFactory.create("BOSTON_HOUSING_APPLY_SVM",
                          NO_METADATA);

// Create physical attribute to flag "case id"
PhysicalAttribute pa =
      m_paFactory.create("ID",
                         AttributeDataType.integerType,
                         PhysicalAttributeRole.caseId );

applyData.addAttribute( pa );

m_dmeConn.saveObject( "svmrApplyData_jdm",
                       applyData, REPLACE );
```

Create Apply Settings



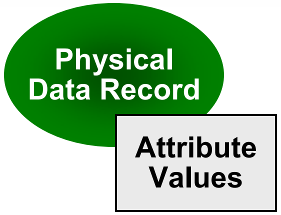
```
// Create default apply settings
RegressionApplySettings regrAS =
      m_applySettingsFactory.create();

// Specify to output the AGE attribute
Map map = new HashMap ();
map.put ("AGE", "AGE1");  // source & destination attr names
regrAS.setSourceDestinationMap (map);

m_dmeConn.saveObject( "svmrApplySettings_jdm",
                       regrAS, REPLACE);
```

Create and Execute the Apply Task



```
// Create the apply task
DataSetApplyTask applyTask = m_dsApplyFactory.create(
      “svmrApplyData_jdm",          // apply data
      "svmrModel_jdm",              // model used for scoring
      "svmrApplySettings_jdm",      // apply settings
      "SVMR_APPLY_OUTPUT_JDM");     // Output: score results

// Save the apply task with a name
m_dmeConn.saveObject("svmrApplyTask_jdm”,
                     applyTask, REPLACE);

// Execute as before
```

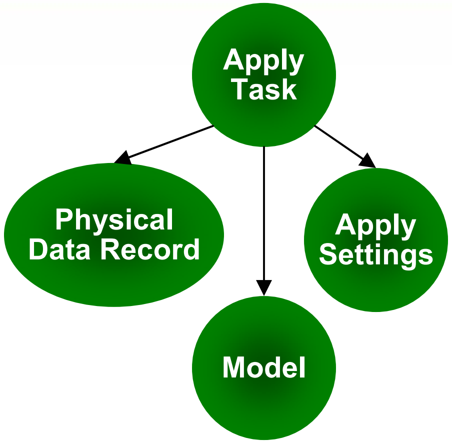
Real-time Scoring


Physical Data Record

Attribute Values

```
// Get the model and its signature
RegressionModel model = (RegressionModel)
       m_dmeConn.retrieveObject("svmrModel_jdm",
                                NamedObject.model);
ModelSignature modelSignature = model.getSignature();

// Create and populate the input record
PhysicalDataRecord applyInputRecord =
       m_pdrFactory.create(modelSignature);

// Prepare data as needed, e.g., normalize (not shown)
applyInputRecord.setValue("CRIM",    new Double(0.006));
applyInputRecord.setValue("ZN",      new Integer(18));
applyInputRecord.setValue("INDUS",   new Double(2.31));
// ...
```

Create the Apply Task



```
// Create the apply task
RecordApplyTask applyTask =
   m_recApplyTaskFactory.create(applyInputRecord,
                                "svmrModel_jdm",
                                "svmrApplySettings_jdm");

// Execute the task synchronously
ExecutionStatus recExecStatus =
   m_dmeConn.execute(applyTask, BLOCK_UNTIL_COMPLETION);

// Check task status after completion as before
```

Retrieve the Predicted House Value

```
// Get output Record
PhysicalDataRecord applyOutputRecord =
       applyTask.getOutputRecord();

// Get prediction value
Double prediction = (Double)
       applyOutputRecord.getValue("PREDICTION");

// Output the house value prediction...
```

DEMO

Execute the Java code

Agenda

Exploring Data Mining

Killer Apps for Data Mining

Building a Data Mining Application

JSR 73 and JSR 247

JDM

Java Community Process

- Open, pure Java technology, multi-vendor standard
- Representative set of techniques and algorithms
- Extensible framework
- *A la carte* conformance
- Novice and expert support
- XML Schema representation for objects
- Web Services interface

JSR 73 Final Specification approved August 2004

JSR 247 Public Review Draft approved December 2006

JDM Expert Group Companies

- BEA Systems
- Computer Associates
- Corporate Intellect
- E.piphany (JDM 2.0)
- Fair Isaac
- Hyperion Solutions
- IBM
- KXEN*
- Oracle*
- SAP
- SAS Institute
- SPSS, Inc.
- Strategic Analytics
- Sun Microsystems

*** Produced JDM implementations**

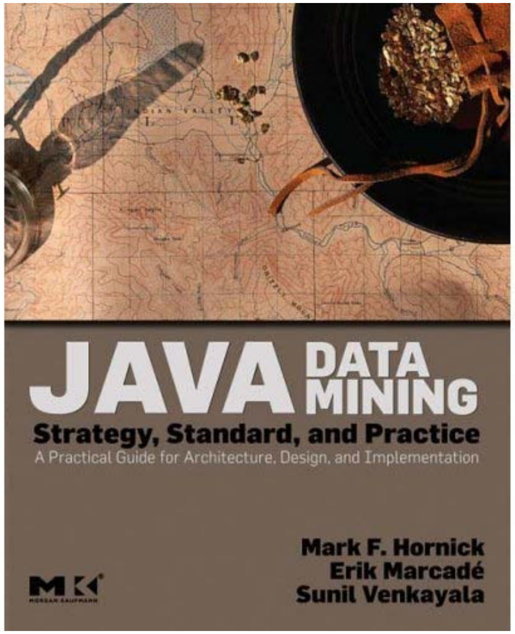
JSR 247 Features

- Time Series
- Anomaly Detection
- Model Comparison
- Transformations
- Multivariate Statistics
- …

Java Data Mining:

Strategy, Standard, and Practice

A Practical Guide for Architecture, Design, and Implementation


JAVA DATA MINING
Strategy, Standard, and Practice
A Practical Guide for Architecture, Design, and Implementation
Mark F. Hornick
Erik Marcadé
Sunil Venkayala
MK

Mark F. Hornick, Oracle

Erik Marcade, KXEN

Sunil Venkayala, Oracle

For More Information

- Download the JDM specifications
  - jcp.org/en/jsr/detail?id=73
  - jcp.org/en/jsr/detail?id=247
- Java.net
  - datamining.dev.java.net and discussion forum
- Try out JDM today
  - oracle.com/technology/products/bi/odm/index.html
  - JDeveloper addin oracle.com/technology/products/bi/odm/odm_jdev_extension.html
  - kxen.com/products/analytic_framework/apis.php

Summary

- Data Mining Demystified
- Data mining technology enables advanced applications
- JDM enables building advanced Java applications

Q&A

Mark Hornick
mark.hornick@oracle.com

Killer Apps:
Data Mining Demystified

Mark Hornick

JSR 73/JSR 247 Specification Lead
Senior Manager
Oracle Corporation
www.oracle.com

TS-1262