Demystifying Data Mining

Carol Brennan Baldan
(Independent)

New York City Metro Area Group Meeting
September 29th, 2005

Agenda

- Goal: Answer questions “What is data mining?” and “How can I get started?”
- Definition of “data mining”
  - Identify the types of business questions that data mining can address
  - Business issues to consider when choosing a data mining model
- Discussion of the structure of an inquiry using data mining
- Examples showing the uses of data mining
  - Business focus
- Overview of Oracle’s data mining solution

What is Data Mining?

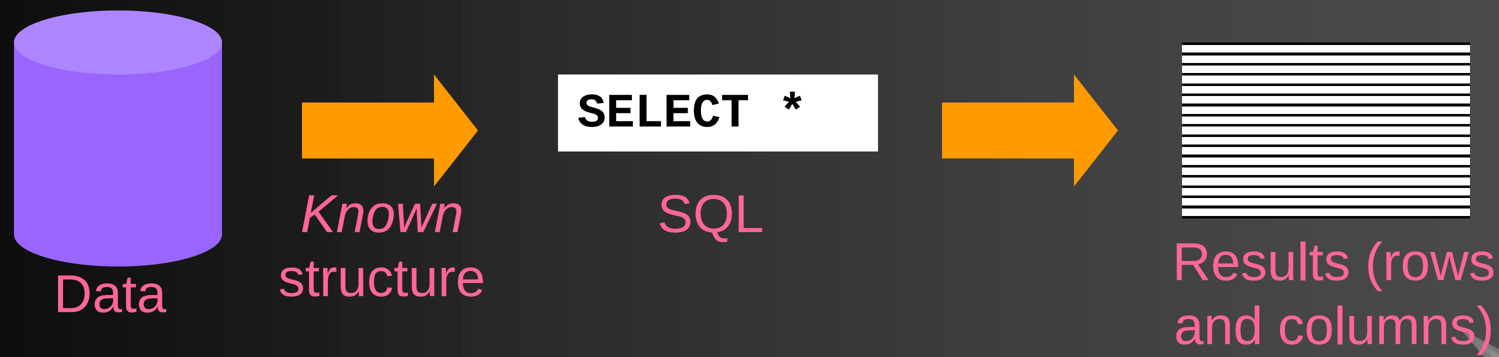
- “Mining” implies a quasi-random search through large quantities of data
  - *This is misleading and over-simplified*
- Data Mining actually involves building a model and using it to make decisions:
  - Use software to build a model
  - The model is based on the structure of our available (i.e. historical) data
  - Then, apply the model to new data to “predict the future”
    - With *many* caveats…

Decision Support: Two Viewpoints

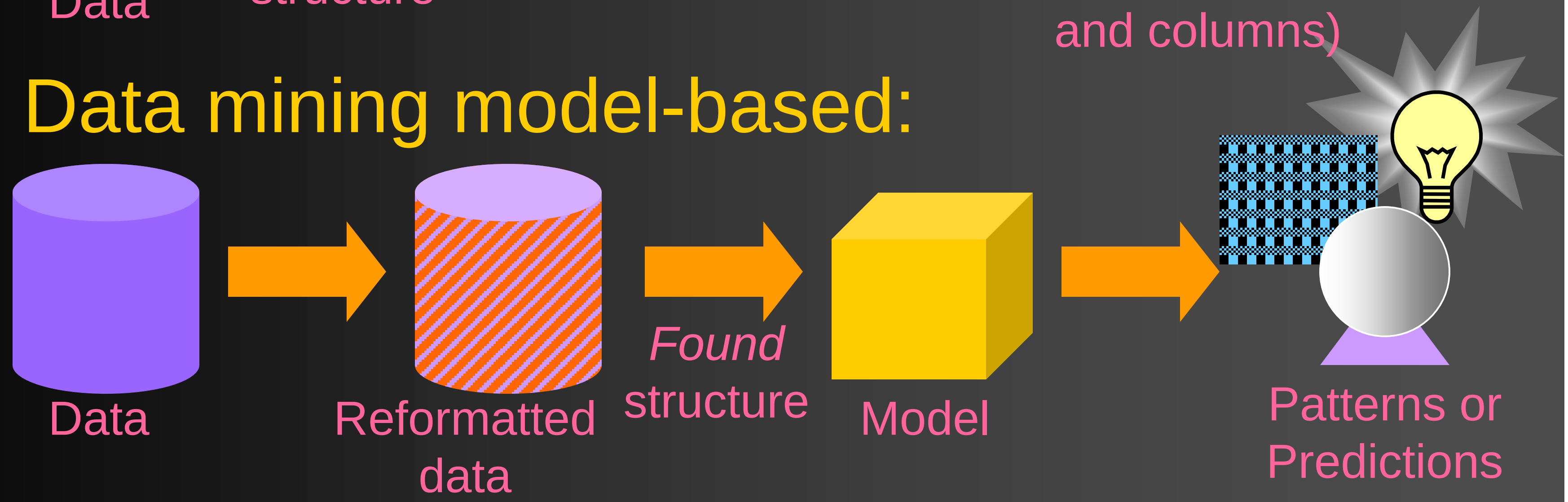
- Normally, we use the known structure of our database to capture data using queries
- With data mining, we build a model based on patterns or structures that our modeling process finds *within* the data

Decision Support: Two Viewpoints

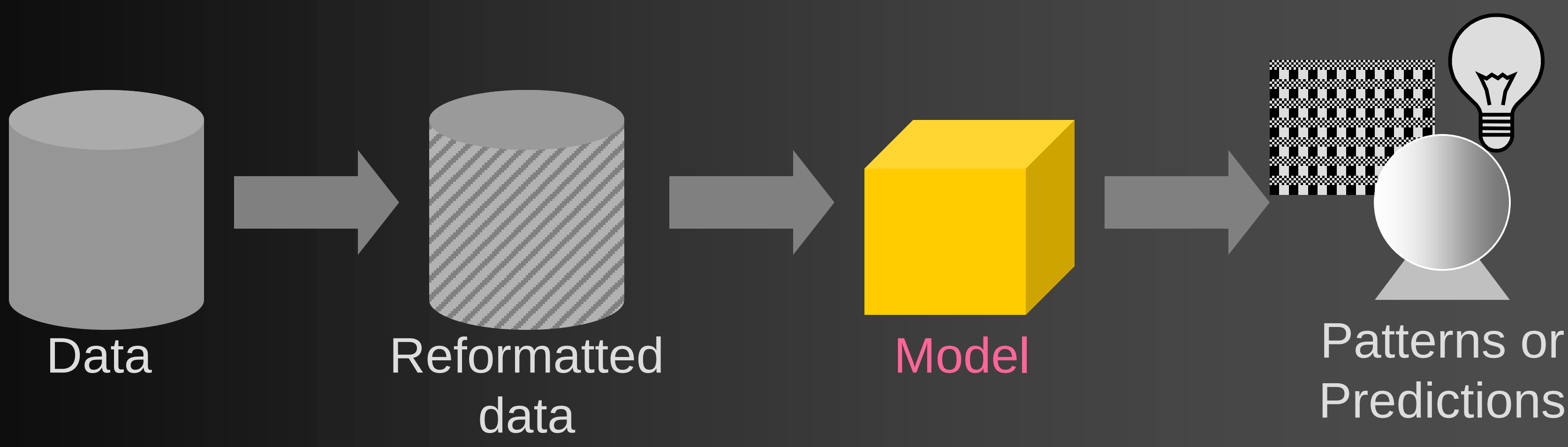
- Traditional:



- Data mining model-based:



Using the Model



The Model: Defined

- “Black box” that produces an answer to a question based on data fed into it
- Example: suppose we have built a model for answering the question “Should our company issue a credit card to this applicant?”

Age: 40

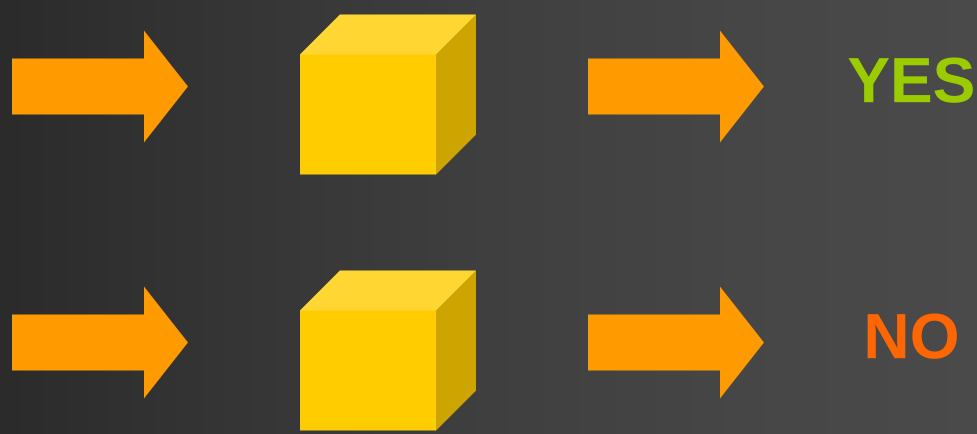
Income: $100,000+/yr

Bank account balance: $10,000

Age: 7

Income: $100/yr

Bank account balance: NULL



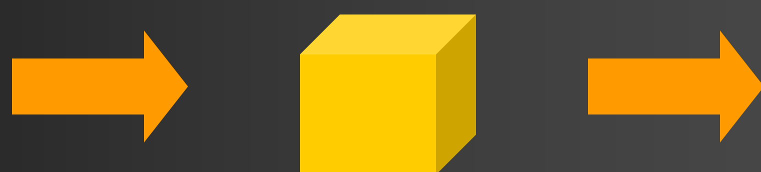
Using The Model: Example

- Reconsider our earlier example: “Should our company issue a credit card to this applicant?”
  - Use age, income, and bank account balance to determine whether or not we should issue credit

Age: 40

Income: $100,000+/yr

Bank account balance: $10,000



YES

Using The Model: Example

- Assume the following:
  - To date, we have issued thousands of credit cards (or more…)
  - For each account, we know the age, income, and bank account balance provided to us when the individual applied for the credit card
  - We also know, for each account, if the card holder repeatedly made late payments, defaulted, went bankrupt, required legal action, attempted fraud, etc.
    - In hindsight, knowing what we now know, which applicants should we *not* have issued credit to?

Using The Model: Example

- Put together historical data:

Age	Annual Income	Bank Acct Balance	Desirable? Yes/No
31	24000	1200	Yes
25	NULL	400	Yes
19	3000	13000	Yes
35	38000	NULL	No
67	28000	50000	No
NULL	97000	700	Yes

- Build the model using this data
  - Conceptually, the model “learns” which factors are indicators of an applicant’s desirability

Using The Model: Example

- Now, consider our new applicants:

Age	Annual Income	Bank Acct Balance	Desirable? Yes/No
NULL	32000	450	?
72	67000	1700	?
21	NULL	NULL	?
37	78000	250	?
28	NULL	600	?
26	12000	1350	?

Using The Model: Example

- We apply the model to this new data to determine the “desirability” value:

Age	Annual Income	Bank Acct Balance	Desirable? Yes/No
NULL	32000	450	Yes
72	67000	1700	Yes
21	NULL	NULL	No
37	78000	250	Yes
28	NULL	600	No
26	12000	1350	Yes

- The model interprets the values of the three known fields and “decides” whether each applicant is desirable

Using The Model: Example

- Can we be certain that the desirable applicants will always pay on time?
  - NO. But… data on our past experience indicates that they are *more likely than average* to always pay on time. This is still useful for our business.
- Can we keep using the same model indefinitely?
  - NO. We will continually open new accounts and get new data on desirability of account holders. Therefore, the body of our historical knowledge is always changing.

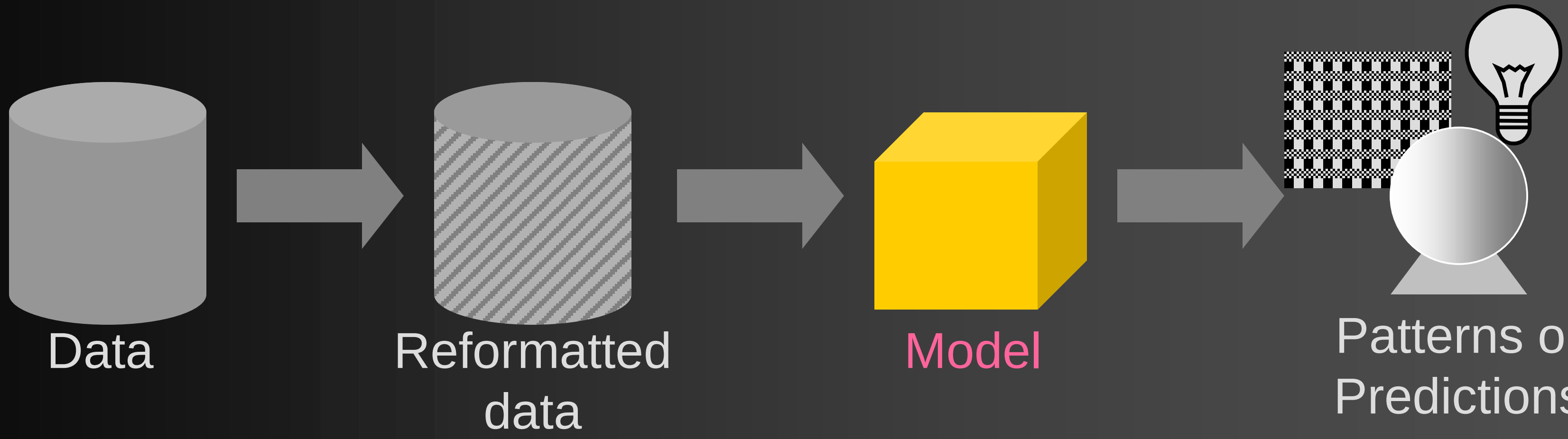
Building The Model

- Two distinct steps:
  - Generate and test the model using one set of data
  - Validate the model on another data set
- If the model is tuned too finely to the data set used to generate/test it, then the validation step will reveal this
  - This phenomenon is called “overfitting”
  - Likely if the original data set has too few data points or is not representative of “real” data

Building The Model

- Not adequate to just use two random samples from the same data pool – statistical tests needed
  - Commonly, we partition by time: Use historical data to build the model and more recent data to validate it
    - But, we need to be careful to not include data that is too old to be relevant in either data set…

Choosing a Model

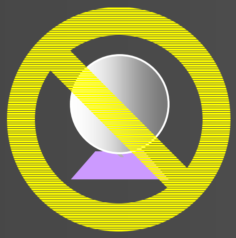


The Model: Considerations

- There are many types of data mining models (probably hundreds!)
  – Vary by product
  – Most products offer a choice of several models
- Which model is "the best"?
  – *It depends!*
- Clarification on terminology:
  – "Model" can refer to either the algorithm for building our decision-making "box" OR the specific framework created using our own data

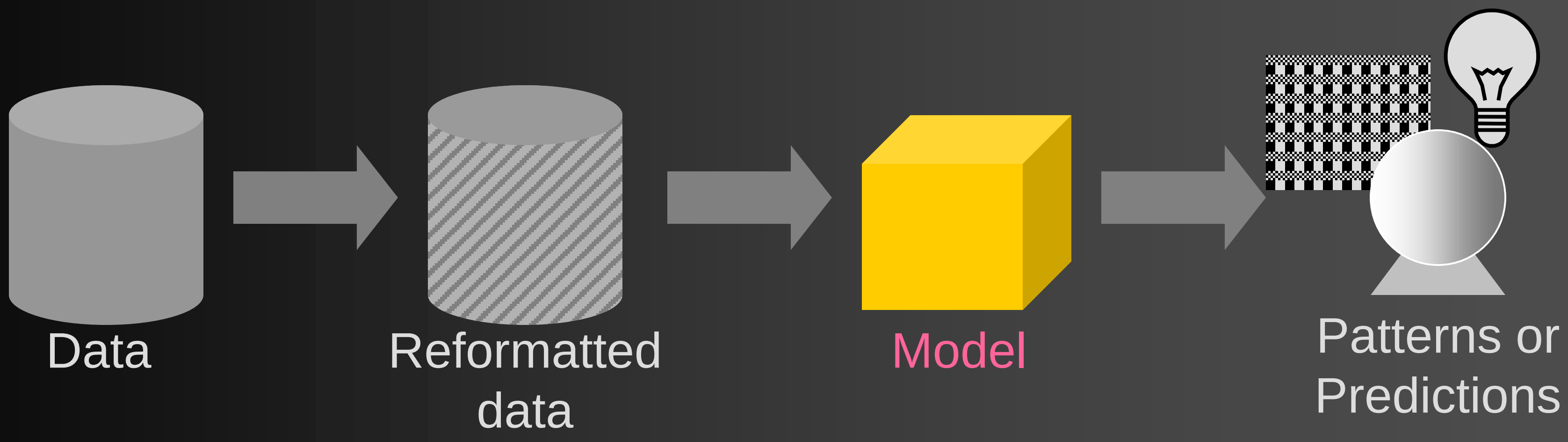
The Model: Considerations

- Factors that will influence your choice of model:
  - **Accuracy**



    - *Note:* No model can achieve 100% accuracy
    - But, 70% guess is better than 50/50…
  - **Transparency**: A model is most useful when business users understand what it does
    - Consider the sophistication of users, training
  - **Tolerance for Sparse or “Noisy” Data**
    - Assess your ability to capture complete and correct data, then choose your model accordingly
  - Others…

Some Common Data Mining Models



Supervised vs. Unsupervised Models

- **Predictive** Models (i.e. **Supervised Learning** Models):
  - Used to predict a value
  - "Supervised" because we specify one value (field) to predict by using the other available values (fields)
- **Descriptive** Models (i.e. **Unsupervised Learning** Models):
  - Used to find intrinsic patterns in data
  - "Unsupervised" because we do not specify any value to predict; we let the model find patterns in the data

Common Models

- Naïve Bayes (Predictive)
- Decision Trees (Descriptive)
- Association Rules (Descriptive)
  - These three models are common to many data mining products and are conceptually less complicated than many other models
- *Not an exhaustive list*

Naïve Bayes Model

- Uses probabilities to determine which of several "classes" a single data point belongs to
  - In our previous example, we had two classes: "Desirable credit card account holder" and "Undesirable credit card account holder"
  - Can have any finite number of classes
  - Based on Bayes' Rule (from statistics)

Naïve Bayes Model

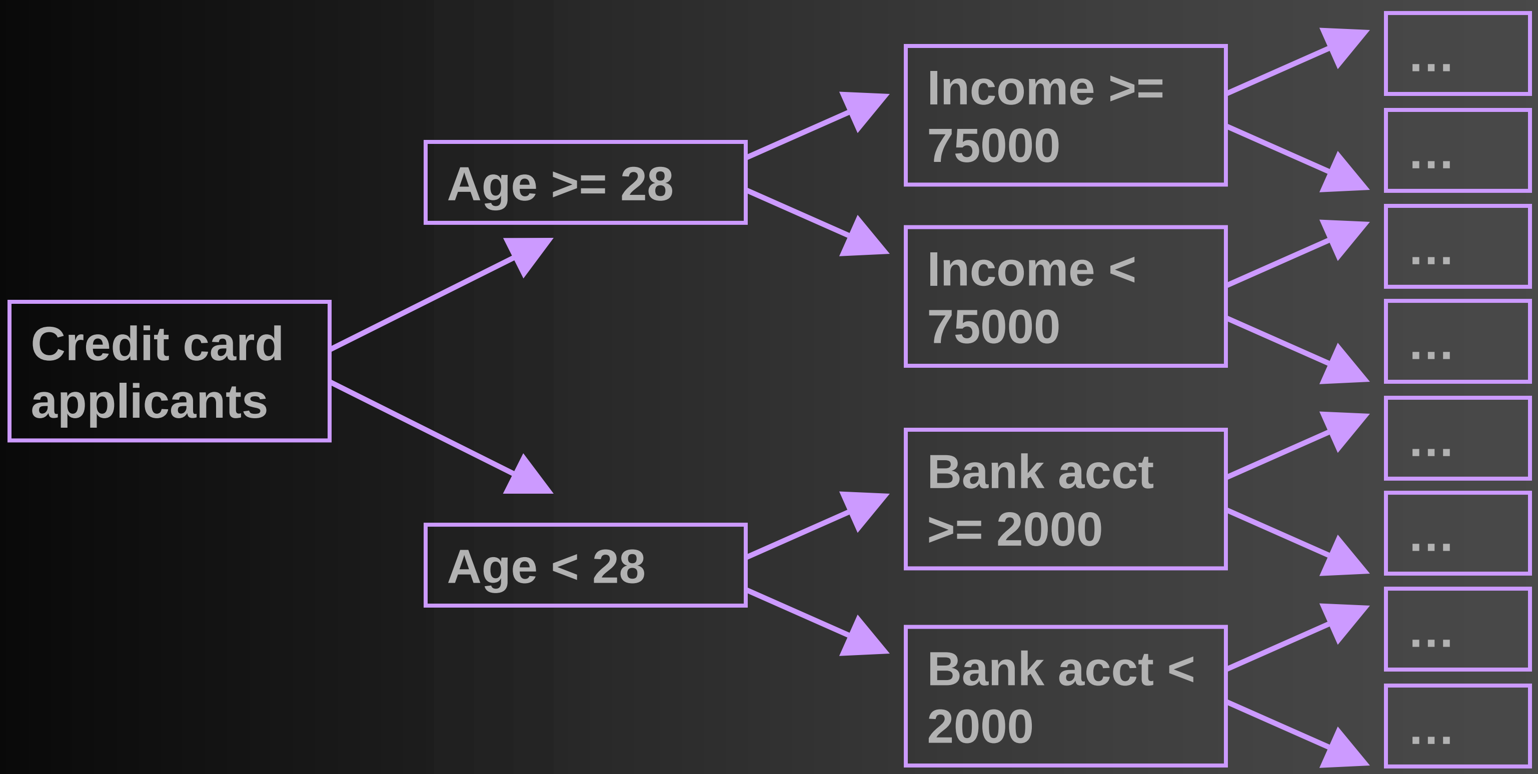
- Model is “naïve” because it assumes that the value of each attribute is independent of the values of other attributes of data points within the same class
  - Not an appropriate model to use if we know that this is not the case
- Will show a specific example using this model later…

Decision Trees

- Sometimes referred to as “rule induction”
- Model specifies series of data “clusters”
  - Data points within clusters are similar (i.e. variance is minimized)
  - Differences between clusters are maximized
  - Determined using statistical methods
- We can then deduce rules for optimally separating data points into clusters

Decision Trees

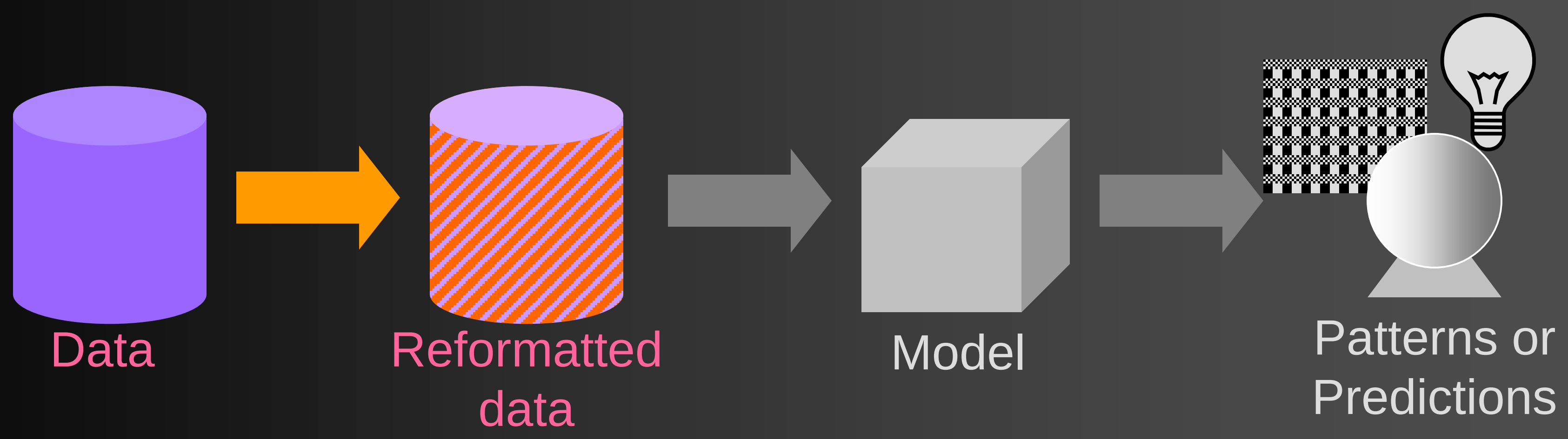
- One possible decision tree, built from our credit-card application data:



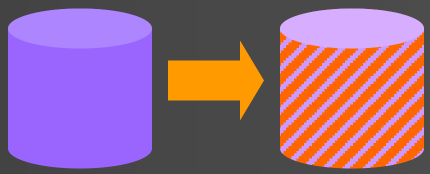
Association Rules

- Discovery technique (not predictive)
- Data set consists of transactions that each contain a set of items
  - Classic example: Items bought by one shopper at a supermarket
- Goal is to find items that occur together
  - For example, hot dogs and hot dog buns

Reformatting Data

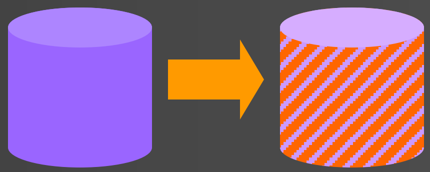


Reformatting Data



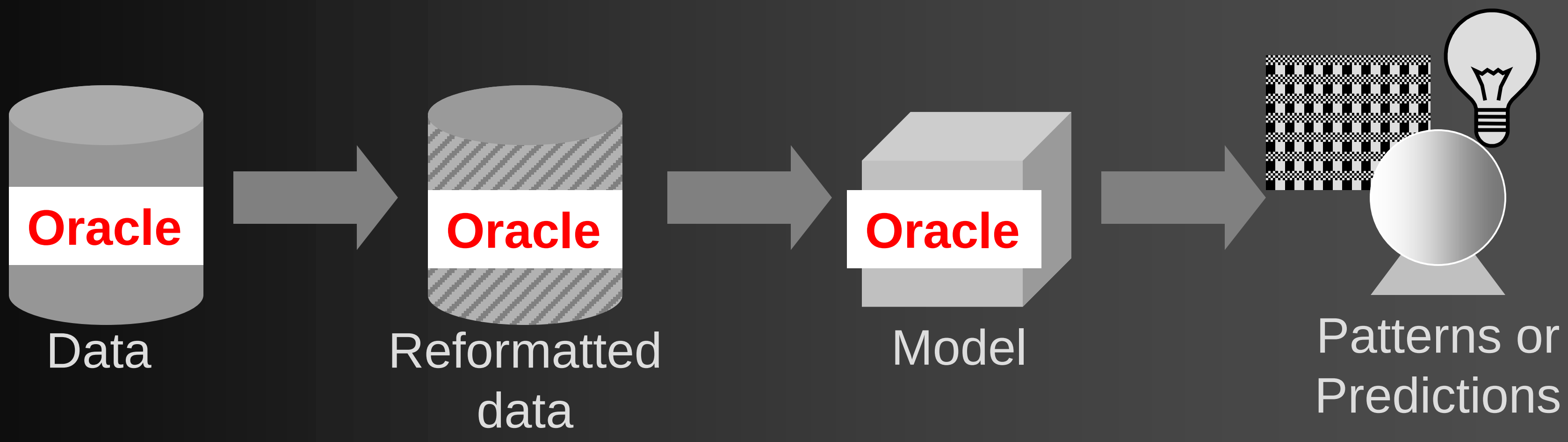
- Data may need to be reformatted or transformed before running a data mining algorithm against it
- Depends on several factors:
  - The product being used
  - The type of model being built
  - The business objective, i.e. the question you are attempting to answer

Reformatting Data



- For example, data mining products often require that data be stored in one “table”, with one row per data point
  - In predictive models, model attempts to predict one column value based on the values of some or all other columns (as seen in our earlier example)
- The situation sometimes calls for **discretization** of values
- New fields may need to be derived from those in the data, if the derived value is relevant to the question

Oracle Data Mining (ODM)



Oracle Data Mining (ODM)

- When installed, encapsulates data mining functions within the database
  - Data, model, and results all contained within the Oracle database
  - Included in Enterprise Edition

Oracle Data Mining (ODM)

- Two interfaces:
  1. ODM Java API
  2. DBMS_DATA_MINING
- Actually two separate products – *not* interoperable
- Different models available within each

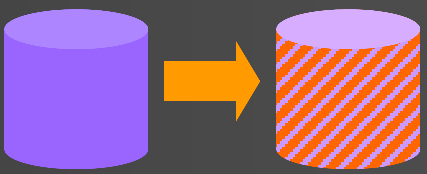
ODM Models

- Predictive:
  - **Classification Models**: Divide items into classes, generate rules for classifying items
    - Includes Naïve Bayes
  - **Regression Models**: Approximate and forecast continuous values
  - **Attribute Importance Models**: Identify attributes that carry the most “weight” in predicting the target value
    - Available within Java API only

ODM Models

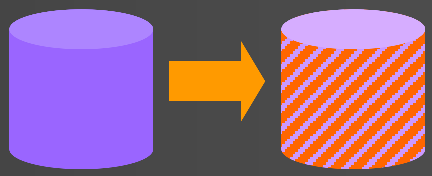
- Descriptive:
  - **Clustering Models**: Identify "groupings" within the data
  - **Association Rules**: Identify values that often occur together
    - "Market basket"
  - **Feature Extraction Models**: Identify features that are combinations of other values

Data Preparation in ODM



- Java Interface works with prepared or unprepared data
- DBMS_DATA_MINING only works with prepared data
- Some models require “binning” (discretization) of variables
  - i.e. specify a continuous value as belonging to one of *N* “bins”

Data Preparation in ODM



- Only works with specific datatypes: VARCHAR2, CHAR, NUMBER, CLOB, BLOB, etc.
- Must convert DATEs to VARCHAR2 or NUMBER, depending on the meaning of the value
- May need to normalize values
  - Perform a conversion of a value such that the result follows the standard normal curve

Example Revisited: ODM

- Question: "Which applicants should we issue credit cards to?"
- Steps (using DBMS_DATA_MINING):
  - 1. Choose an appropriate mining algorithm for the problem. (Assume we have chosen Naïve Bayes.)
  - 2. Identify data for building/testing and validating the model
    - Oracle refers to the validation step as "scoring"
  - 3. Prepare the data using SQL, PL/SQL, third-party tools, or the DBMS_DATA_MINING package
    - In our case, we will need to discretize values – we can do this with DBMS_DATA_MINING or by generating new tables

Example Revisited: ODM

- 4. Build a settings table
  - We can choose the name, but it must have this structure:

    (*setting_name* VARCHAR2(30),

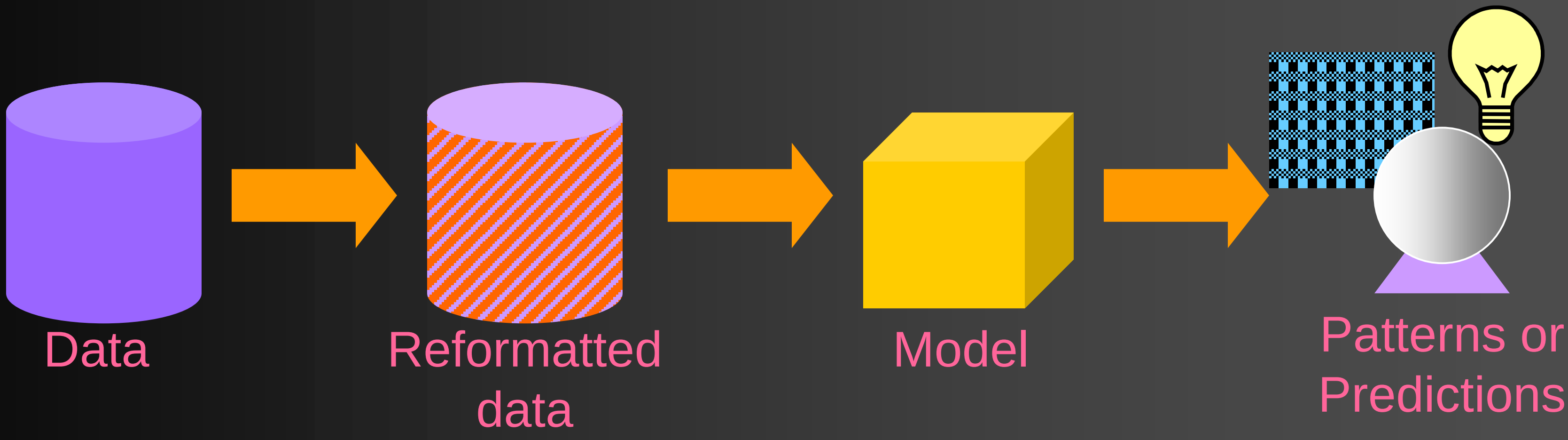
    setting_value VARCHAR2(128))
  - In our case, we will insert this setting:

    (algo_name, algo_naive_bayes)
  - Other settings available which are specific to the Naïve Bayes model
    - Assume here that we are accepting defaults

Example Revisited: ODM

- 5. Create model using DBMS_DATA_MINING.CREATE_MODEL
- 6. Create a results table using DBMS_DATA_MINING.APPLY
- 7. Test/validate with new data using DBMS_DATA_MINING.COMPUTE (specific to Naïve Bayes and similar models)
- 8. Analyze tests/validations using statistical methods

Further Information



Additional Notes: ODM

- Other procedures included in DBMS_DATA_MINING
- Can mine “wide” data (i.e. records that exceed Oracle’s column limit) with “multi-record case format” (provided)

Data Mining: General Issues

- Technical
  - Management of large data sets (e.g. data warehouse, “wide” data)
  - Scalability, flexibility, speed
  - Development resources
- Organizational
  - Availability of technical/model expertise
  - Confidence in imprecise “answers”
  - Potentially steep learning curve

Other Data Mining Products

- SAS Data Mining
  - SAS Enterprise Miner and SAS Text Miner
  - http://www.sas.com/technologies/analytics/datamining/index.html
- WEKA (Waikato Environment for Knowledge Analysis)
  - http://www.cs.waikato.ac.nz/~ml/weka/index.html
  - Open source (GNU General Public License)
- CART (Classification and Regression Trees)
  - http://www.salford-systems.com/cart.php
  - Commercial product, 30-day evaluation available
- Many others…

Suggested Reading

- Seven Methods for Transforming Corporate Data into Business Intelligence, Vasant Dhar and Roger Stein
- Oracle 9.0.1/9.2/10g Data Mining Documentation
- Many Web sites…

Questions?

Thanks!

Carol Brennan Baldan

carol.baldan@patmedia.net or
carolbv2@yahoo.com