Finding Rare Needles in the Haystack

Predictive Modeling with Unbalanced Data

Paul Bradley, *Chief Scientist, MethodCare, Inc.*

24 HOURS OF PASS

Global Sponsor: Microsoft

Overview

- Unbalanced Data Exists and Lives
- Solutions and Approaches
  - Algorithm Parameter Tuning
  - Data Sampling
  - Different Costs for Erroneous Predictions
- Healthcare Readmissions Application

Unbalanced Data

- "In life it is often the rare objects that are most interesting…"
  - G. M. Weiss. Mining with Rarity: A Unifying Framework. SIGKDD Explorations, Vol. 6-1, pp. 7 – 19.
- Rare objects are typically more difficult to find… most data mining algorithms have a great deal of difficulty dealing with rarity.

- Unbalanced Data ⇔ Rare Cases
  - The object of interest (to be predicted) is very infrequent w.r.t. alternate objects
  - Object of interest = Class 1; Alternate objects = Class 0
  - Frequency of Class 1 in the data <<< Frequency of Class 0 in the data.

Examples

- Identify fraudulent credit card transactions
  - Proportionally few transactions are fraudulent
- Predicting telecommunications equipment failures
  - Few examples of actual failures
- Detecting oil spills from satellite imagery
  - 41 of 937 satellite images contain oil slicks
- Identifying patients likely to readmit for a given diagnosis
  - Relatively few readmits vs. non-readmits

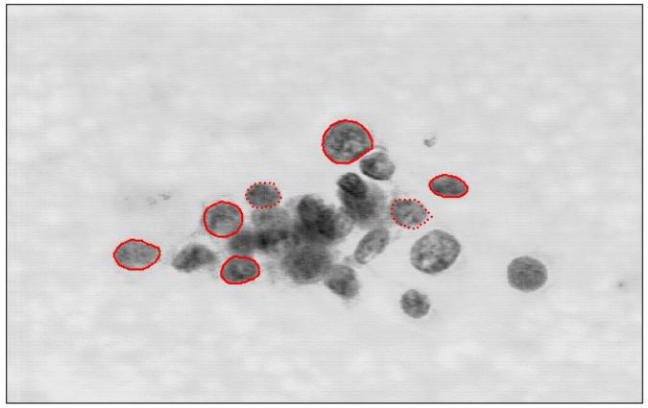
What Makes Unbalanced Data Hard?

- Most predictive modeling algorithms maximize accuracy
  - Assume 2% of your dataset consist of the interesting object (rare)
  - 98% consists of the uninteresting object
  - Always predicting “not interesting” => 98% accuracy rate
    - Good accuracy rate!
    - Very uninteresting model – never will identify the item of interest.

- Now what can you do?
  - Adjust algorithm parameters to (attempt) to identify the rare cases
  - Sampling techniques to create more balanced dataset
    - Proportion of cases that are interesting is large enough so they are identified
  - Predict to minimize *cost* vs. maximize accuracy

Example Dataset Attributes

- Measurements from image for FNA
  - 10 features derived
  - Average, Standard Error, Extreme Value
    - 30 features



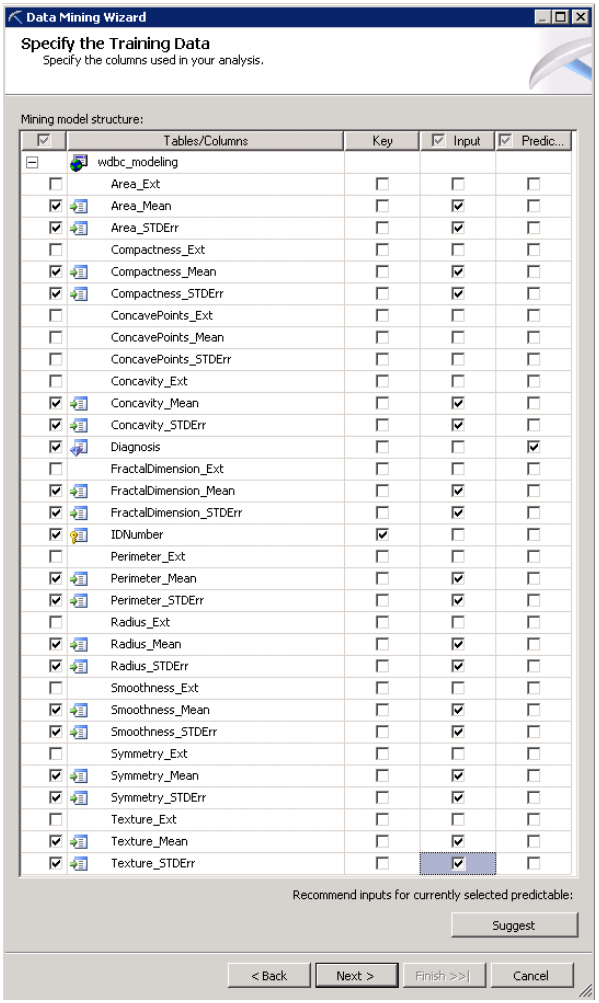
- Diagnosis confirmed by pathology
- Example: Ignoring extreme values and concave points attributes
- Class Distribution
  - Original Dataset:
    - Benign: 62.7%; Malignant: 37.3%
  - Modeling Dataset
    - Benign: 93.2%; Malignant: 6.8% (rare class)

Algorithm Parameter Tuning

- Build Mining Structure over Example Dataset
  - Brief overview of process using Visual Studio Analysis Services Project
- Build Mining Models
  - Microsoft Decision Trees
  - Similar concepts apply for the other DM algorithms in SSAS
    - Microsoft Association Rules, Microsoft Clustering, Logistic Regression, Naïve Bayes, Neural Networks
- Explore
  - Models with default parameters
  - Tuning parameters to identify the rare cases

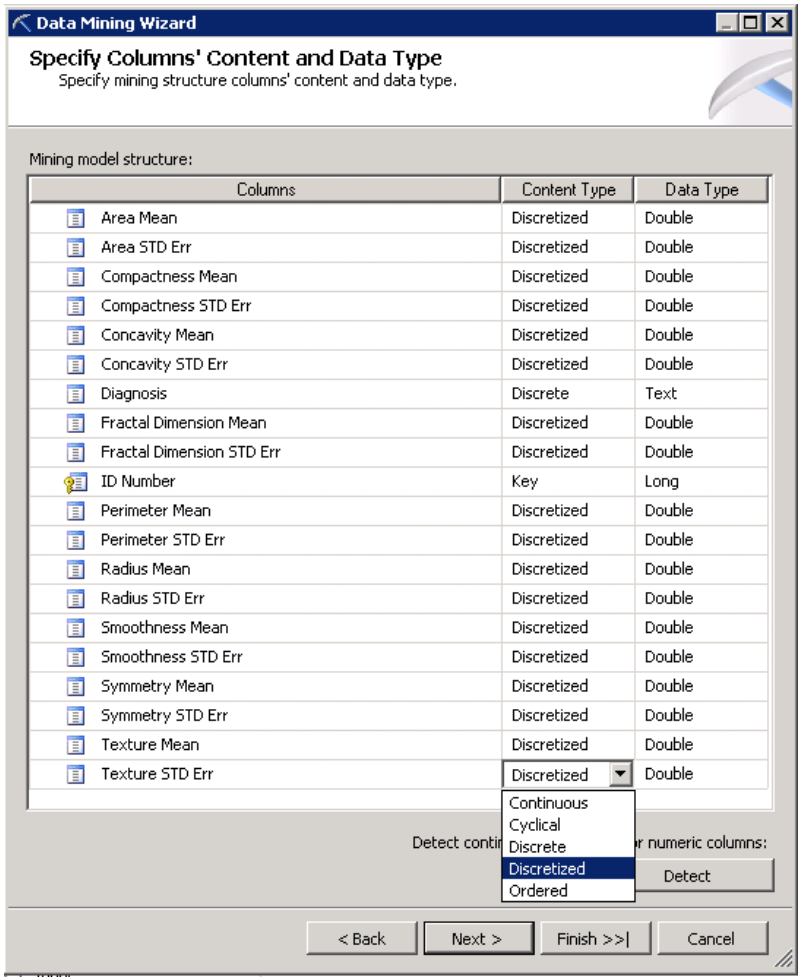
SSAS Mining Structure

- Specify:
  - Case Key
  - Input Attributes
  - Output Attribute



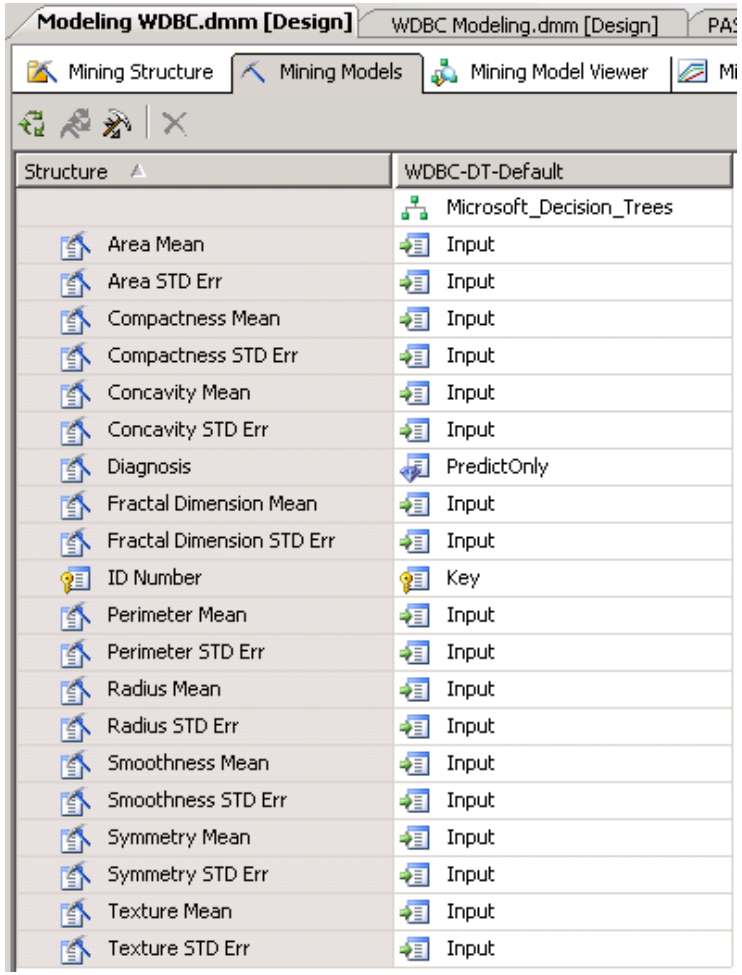
SSAS Mining Structure

- Specify:
  - Column Contents
    - Discretize continuous attributes
  - Data Type



SSAS Mining Structure

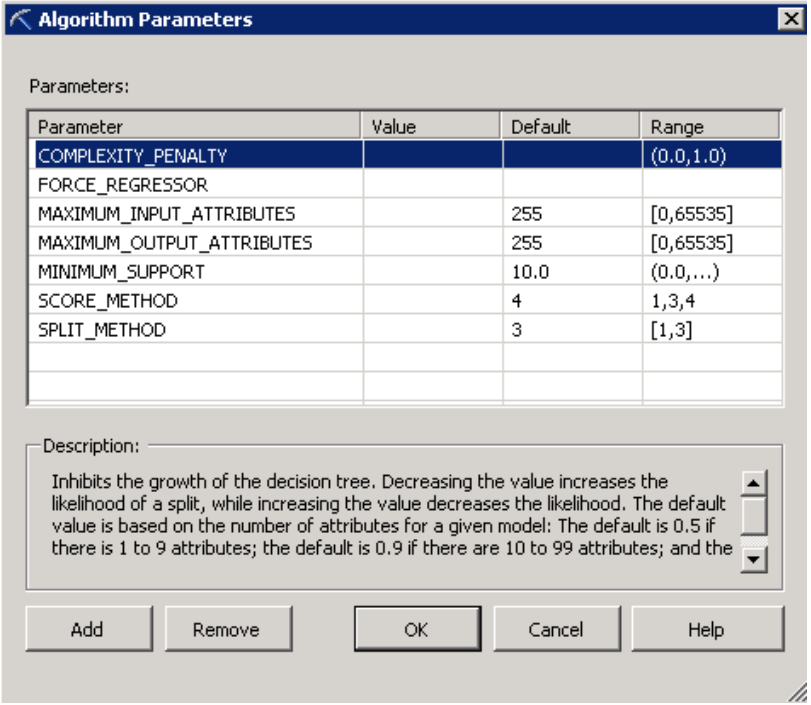
- Define Decision Tree



Structure	WDBC-DT-Default
	Microsoft_Decision_Trees
Area Mean	Input
Area STD Err	Input
Compactness Mean	Input
Compactness STD Err	Input
Concavity Mean	Input
Concavity STD Err	Input
Diagnosis	PredictOnly
Fractal Dimension Mean	Input
Fractal Dimension STD Err	Input
ID Number	Key
Perimeter Mean	Input
Perimeter STD Err	Input
Radius Mean	Input
Radius STD Err	Input
Smoothness Mean	Input
Smoothness STD Err	Input
Symmetry Mean	Input
Symmetry STD Err	Input
Texture Mean	Input
Texture STD Err	Input

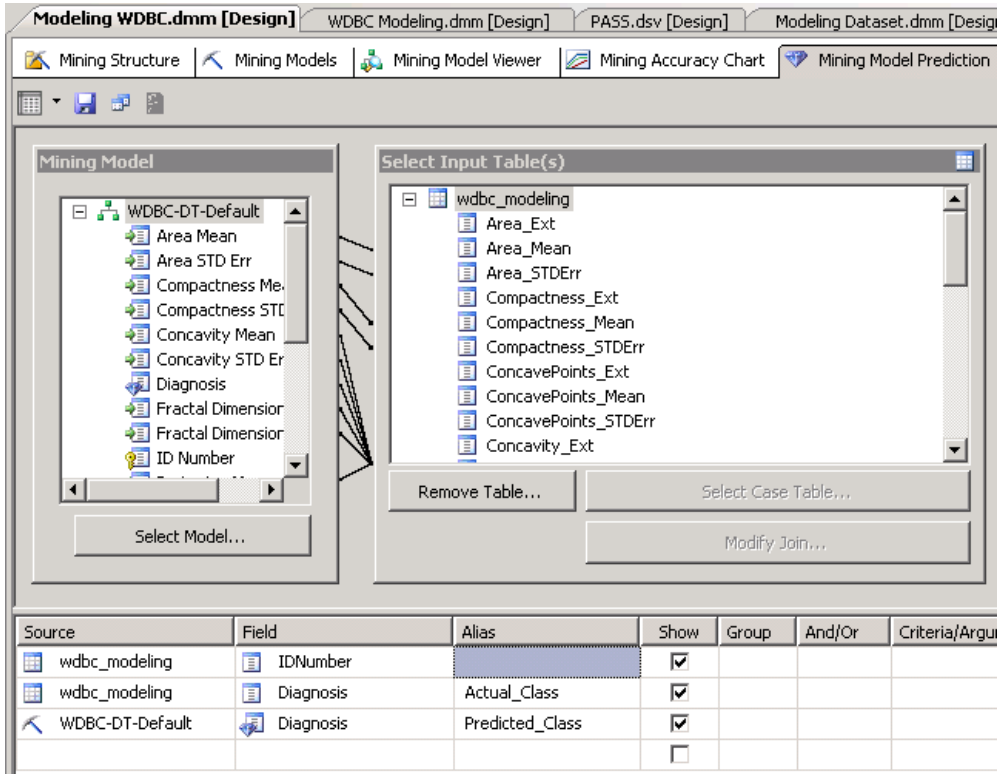
SSAS Mining Models

- Default Decision Tree Parameters



Build Models, Analyze Predictions

- Create table with Actuals and Predictions



Compare Actuals vs Predictions

```
Select     Actual_Class
           , Predicted_Class
           , count(*) as num_Cases
From DT_Default_Predictions
Group by Actual_Class, Predicted_Class
```

	Actual_Class	Predicted_Class	num_cases
1	B	B	357
2	M	B	26

Decision Tree Learning and Rare Cases

- Learning
  - Decision trees iteratively split the data until leaf nodes are found s.t.:
    - The majority of the cases at the leaf node belong to 1 class
    - The node cannot be further split based on tree growth (or pruning) criteria

- How to identify the rare cases
  - Goal
    - Tune parameters so that leaf nodes are found contain a majority of rare cases.
  - How?
    - Generate more data splits

Decision Tree Parameter Tuning

- Important parameters to recognize rare cases
  - COMPLEXITY_PENALTY
    - “Inhibits the growth of the decision tree. Decreasing this value increases the likelihood of a split, while increasing the value decreases the likelihood…”
  - MAXIMUM_INPUT_ATTRIBUTES
    - “Specifies the maximum number of input attributes that the algorithm can handle before invoking features selection. Setting this value to 0 disables feature selection…”
  - MINIMUM_SUPPORT
    - “Specifies the minimum number of cases that a leaf node must contain…”

COMPLEXITY_PENALTY Tuning

- Definition
  - “Inhibits the growth of the decision tree. Decreasing this value increases the likelihood of a split, while increasing the value decreases the likelihood…”

- Decrease COMPLEXITY_PENALTY
  - Will increase the likelihood of a split =>
  - Increase the likelihood of a leaf node containing a majority of rare cases
  - Experimentation required to determine actual value
  - Get more complex, larger trees

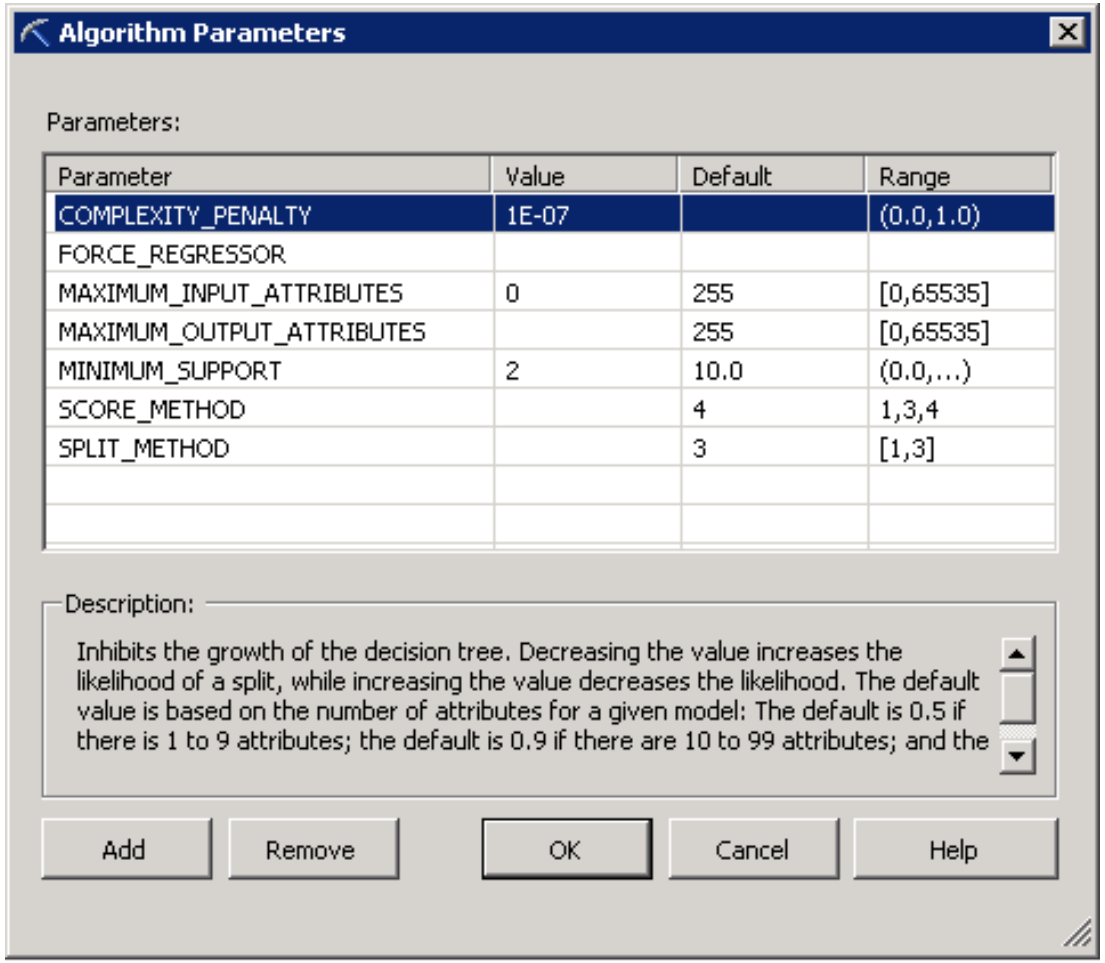
MAXIMUM_INPUT_ATTRIBUTES Tuning

- Definition
  - "Specifies the maximum number of input attributes that the algorithm can handle before invoking feature selection. Setting this value to 0 disables feature selection…"
- Feature Selection
  - Before building the tree, select a subset of data attributes to use for building.
    - Large body of machine learning literature on feature selection
- Increase MAXIMUM_INPUT_ATTRIBUTES
  - Will allow more attributes to be candidates for a split =>
  - Increase the likelihood that a leaf node containing a majority of rare cases

MINIMUM_SUPPORT Tuning

- Definition
  - “Specifies minimum number of cases that a leaf node must contain…”
- Decrease MINIMUM_SUPPORT
  - Will allow the algorithm to continue to split nodes to get to leaf =>
  - Increase the likelihood that a leaf node containing a majority of rare cases

Updated DT Parameters



Updated Parameter Model Build

- Compare Actuals to Predictions

```
Select     Actual_Class
           , Predicted_Class
           , count(*) as num_Cases
From DT_Iter1_Predictions
Group by Actual_Class, Predicted_Class
```

	Actual_Class	Predicted_Class	num_cases
1	B	B	356
2	B	M	1
3	M	B	5
4	M	M	21

Parameter Tuning Summary

- Goal
  - Attempt to build models that identify rare cases by being more specific and complex.

- Example with Microsoft Decision Trees
  - Build large, more complex trees so that leaf nodes have majority of the rare class.
  - Evaluate by comparing actual class values with predicted values

Data Sampling – Background

- Motivation
  - Give more “importance” to the rare cases
  - See:
    - G. Batista, R. Prati, M. Monard. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explorations, Vol. 6, Iss. 1, June 2004.

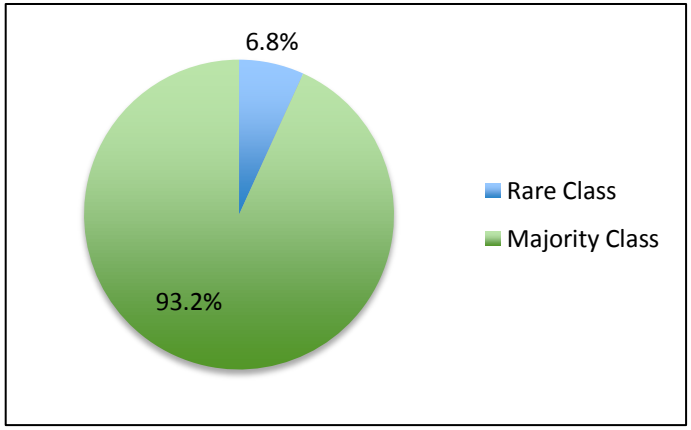
- Skew the class distribution for modeling
  - Create a “training” dataset that includes:
    - All of the cases from the rare class
    - A random sample of cases from the majority class

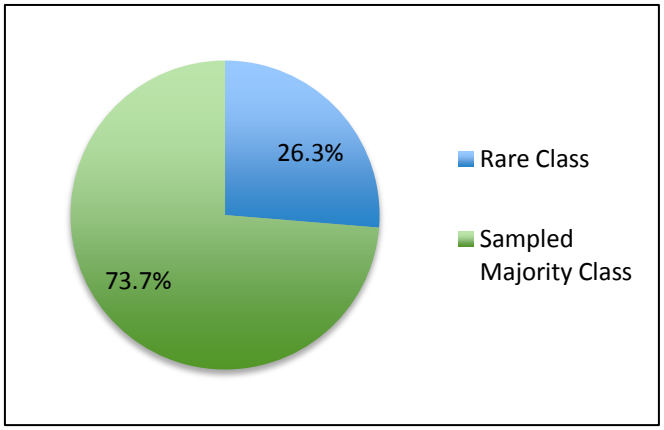
- Use original dataset for evaluation
  - Original class distribution
  - Or test set with original class distribution

Data Sampling – Process

- Create training set:
  - All of the rare cases
  - Random sample of the majority class
  - Training set skewed class distribution
    - Rare class: 20% or more
    - Majority class: 80% or less

- Experiment to determine the “best” distribution





Data Sampling – Implementation

- Random Sampling in SQL Server
  - See:
    - M. De Barros, K. Gidewall. Selecting Rows Randomly from a Large Table. http://msdn.microsoft.com/en-us/library/cc441928.aspx

- Example

```
select  *
into    WDBC_Sample
from    (
            select  *
            from    wdbc_modeling
            where   Diagnosis = 'M'

            union

            select  *
            from    wdbc_modeling
            where   Diagnosis = 'B'
            and     abs(10000.0*(rand(IDNumber)) - round(10000.0*(rand(IDNumber)),0)) < 0.1
        ) T
```

Modeling and Scoring

- Build model over the sampled, skewed distribution
- Compare actuals vs. predictions over original non-skewed distribution

	Actual_Class	Predicted_Class	num_cases
1	B	B	325
2	B	M	32
3	M	B	8
4	M	M	18

Data Sampling Tuning Summary

- Goal
  - Skew the class distribution to give the rare class more “importance”

- Example
  - Construct “training” dataset with skewed class distribution
    - E.g. Rare class: 20% or more; Majority class: 80% or less
  - Build models over the sampled dataset with skewed distribution
  - Evaluate by comparing actual class values with predicted values over original dataset
    - Or over test set with original class distribution

Different Costs for Erroneous Predictions

- Goal
  - Make the models “aware” of different costs of erroneous predictions
    - Cost of predicting “benign” when actually “malignant” > Cost of predicting “malignant” when actually “benign”
    - Want model to minimize overall costs of erroneous predictions
- Problem
  - Predictive modeling algorithms in SSAS all maximize accuracy
    - Equivalent to minimizing costs when costs of errors are the same
- One Solution – MetaCost
  - Create a training set with altered class labels to account for erroneous prediction costs
  - Build model over altered class labels
  - Evaluate over original dataset (or test set with original class distribution)

MetaCost

- P. Domingos. MetaCost: A General Method for Making Classifiers Cost-Sensitive.
  - In *Proc. 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pp. 155-164. ACM Press. 1999.

- Goal:
  - Using *accurate* probability estimates of likelihood of class = 'M'
  - Update class labels of each case to minimize overall *risk*
    - Minimizes the (probability of class = 'M')*(cost of predicting class = 'M')

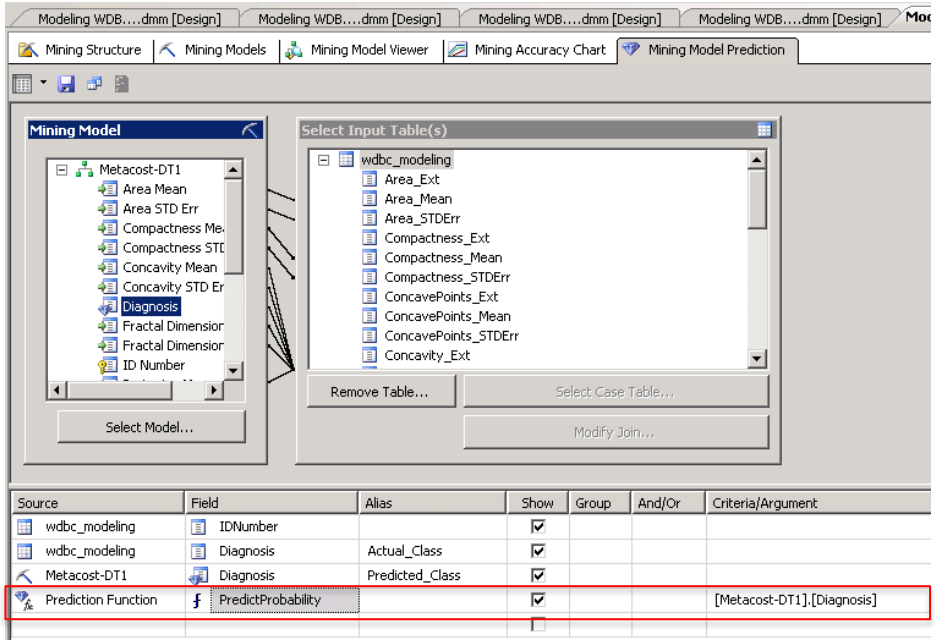
- Implementation
  - To get accurate probabilities, build multiple models over data samples
  - Average the probabilities produced over each sample

MetaCost Example

- Costs
  - Cost of predicting ‘benign’ when case is actually ‘malignant’: 10
  - Cost of predicting ‘malignant’ when case is actually ‘benign’: 1
- Probability estimates
  - Example: Take 5 random samples from original dataset
    - Sample 67% of the cases
  - Build model over the sample
  - Score the entire dataset, getting probability of ‘malignant’ for each case
  - Average the probability of ‘malignant’ for each case to get more accurate estimates.
- Altered class labels
  - Taking into account costs, generate new class labels over the whole dataset

Predictions – PredictProbability

- Build models over each sample
- Make predictions over the entire dataset
  - Include probability of the prediction



Compute P(Malignant) over each sample

- If predicted value = ‘M’, then PredictProbability is P(Malignant)
- If predicted value = ‘B’, then (1-PredictProbability) is P(Malignant)

```
select    MC1.IDNumber
        , case
            when MC1.Predicted_Class = 'M' then MC1.[Expression]
            else (1.0 - MC1.[Expression])
          end as Prob_Malignant_1
        , case
            when MC2.Predicted_Class = 'M' then MC2.[Expression]
            else (1.0 - MC2.[Expression])
          end as Prob_Malignant_2
        , case
            when MC3.Predicted_Class = 'M' then MC3.[Expression]
            else (1.0 - MC3.[Expression])
          end as Prob_Malignant_3
        , case
            when MC4.Predicted_Class = 'M' then MC4.[Expression]
            else (1.0 - MC4.[Expression])
          end as Prob_Malignant_4
        , case
            when MC5.Predicted_Class = 'M' then MC5.[Expression]
            else (1.0 - MC5.[Expression])
          end as Prob_Malignant_5
from      WDBC_MetaCost1_Predictions MC1
          inner join WDBC_MetaCost2_Predictions MC2 on MC1.IDNumber = MC2.IDNumber
          inner join WDBC_MetaCost3_Predictions MC3 on MC1.IDNumber = MC3.IDNumber
          inner join WDBC_MetaCost4_Predictions MC4 on MC1.IDNumber = MC4.IDNumber
          inner join WDBC_MetaCost5_Predictions MC5 on MC1.IDNumber = MC5.IDNumber
```

	IDNumber	Prob_Malignant_1	Prob_Malignant_2	Prob_Malignant_3	Prob_Malignant_4	Prob_Malignant_5
1	8913	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286
2	8915	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0410094637223976	0.104991394148021
3	9047	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286
4	85715	0.553571428571429	0.39672131147541	0.445255474452555	0.0410094637223976	0.104991394148021
5	86211	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0410094637223976	0.104991394148021
6	86408	0.0592686002522068	0.39672131147541	0.445255474452555	0.0410094637223976	0.104991394148021
7	86409	0.0592686002522068	0.39672131147541	0.445255474452555	0.0410094637223976	0.104991394148021
8	86561	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286
9	87106	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286
10	87127	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286
11	87164	0.0592686002522068	0.39672131147541	0.445255474452555	0.465648854961832	0.104991394148021
12	87930	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0410094637223976	0.104991394148021
13	89296	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0410094637223976	0.104991394148021
14	89344	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286
15	89346	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286
16	89524	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0410094637223976	0.00798084596967286
17	89813	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0410094637223976	0.104991394148021
18	89827	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0100502512562812	0.00798084596967286

Average the Probabilities

- For each case, average the P(Malignant) values to get better estimate:

	IDNumber	Prob_Malignant_1	Prob_Malignant_2	Prob_Malignant_3	Prob_Malignant_4	Prob_Malignant_5	AvgProb_Malignant
1	8913	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286	0.0201128772485187
2	8915	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0410094637223976	0.104991394148021	0.0588090691075247
3	9047	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286	0.0201128772485187
4	85715	0.553571428571429	0.39672131147541	0.445255474452555	0.0410094637223976	0.104991394148021	0.308309814473962
5	86211	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0410094637223976	0.104991394148021	0.0457068293774115
6	86408	0.0592686002522068	0.39672131147541	0.445255474452555	0.0410094637223976	0.104991394148021	0.209449248810118
7	86409	0.0592686002522068	0.39672131147541	0.445255474452555	0.0410094637223976	0.104991394148021	0.209449248810118
8	86561	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286	0.0201128772485187
9	87106	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286	0.0201128772485187
10	87127	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286	0.0201128772485187
11	87164	0.0592686002522068	0.39672131147541	0.445255474452555	0.465648854961832	0.104991394148021	0.294377127058005
12	87930	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0410094637223976	0.104991394148021	0.0588090691075247
13	89296	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0410094637223976	0.104991394148021	0.0457068293774115
14	89344	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286	0.0201128772485187
15	89346	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0100502512562812	0.00798084596967286	0.0201128772485187
16	89524	0.0592686002522068	0.00798084596967286	0.0152838427947597	0.0410094637223976	0.00798084596967286	0.026304719741742
17	89813	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0410094637223976	0.104991394148021	0.0588090691075247
18	89827	0.0592686002522068	0.0239294710327457	0.0648464163822526	0.0100502512562812	0.00798084596967286	0.0332151169786318

Compute Updated Class Labels

- Compute updated class labels to minimize risk
  - Costs:
    - Cost of predicting ‘Benign’ when actually ‘Malignant’: 10
    - Cost of predicting ‘Malignant’ when actually ‘Benign’: 1
  - Risk of predicting ‘Benign’ when actually ‘Malignant’:
    - (Avg Prob(Benign))*(Cost of predicting ‘Benign’ when actually ‘Malignant’)
  - Risk of predicting ‘Malignant’ when actually ‘Benign’):
    - (Avg Prob(Malignant))*(Cost of predicting ‘Malignant’ when actually ‘Benign’)

  - Choose the class label the minimizes risk:
    - If (Risk of predicting ‘Benign’ when actually ‘Malignant’) < (Risk of predicting ‘Malignant’ when actually ‘Benign’), then label = ‘Benign’
    - Else label = ‘Malignant’

Updated Class Labels – Example

```
select    WDBC_AvgProbs.IDNumber
        , AvgProb_Malignant
        , AvgProb_Malignant*10.0 as Cost_Benign
        , (1.0 - AvgProb_Malignant)*1.0 as Cost_Malignant
        , case
            when  AvgProb_Malignant*10.0 < (1.0 - AvgProb_Malignant)*1.0 then 'B'
            else 'M'
          end as NewDiagnosis
        , Diagnosis
from    WDBC_AvgProbs
        inner join wdbc_modeling on WDBC_AvgProbs.IDNumber = wdbc_modeling.IDNumber
```

	IDNumber	AvgProb_Malignant	Cost_Benign	Cost_Malignant	NewDiagnosis	Diagnosis
1	8913	0.0201128772485187	0.201128772485187	0.979887122751481	B	B
2	8915	0.0588090691075247	0.588090691075247	0.941190930892475	B	B
3	9047	0.0201128772485187	0.201128772485187	0.979887122751481	B	B
4	85715	0.308309814473962	3.08309814473962	0.691690185526038	M	M
5	86211	0.0457068293774115	0.457068293774115	0.954293170622588	B	B
6	86408	0.209449248810118	2.09449248810118	0.790550751189882	M	B
7	86409	0.209449248810118	2.09449248810118	0.790550751189882	M	B
8	86561	0.0201128772485187	0.201128772485187	0.979887122751481	B	B
9	87106	0.0201128772485187	0.201128772485187	0.979887122751481	B	B
10	87127	0.0201128772485187	0.201128772485187	0.979887122751481	B	B
11	87164	0.294377127058005	2.94377127058005	0.705622872941995	M	M
12	87930	0.0588090691075247	0.588090691075247	0.941190930892475	B	B
13	89296	0.0457068293774115	0.457068293774115	0.954293170622588	B	B
14	89344	0.0201128772485187	0.201128772485187	0.979887122751481	B	B
15	89346	0.0201128772485187	0.201128772485187	0.979887122751481	B	B
16	89524	0.026304719741742	0.26304719741742	0.973695280258258	B	B
17	89813	0.0588090691075247	0.588090691075247	0.941190930892475	B	B

Now that we have updated labels…

- Class distribution
  - Original dataset:
    - Benign: 93.2%; Malignant: 6.8%
  - Updated labels:
    - Benign: 80.7%; Malignant: 19.3%

- Build model using new labels to minimize overall *risk*
- Evaluate over original dataset
  - Or test set with original class distribution

	Actual_Class	Predicted_Class	num_cases
1	B	B	302
2	B	M	55
3	M	B	7
4	M	M	19

Example MetaCost for Readmissions

Run	Algorithm	C(0,1)	C(1,0)	TotalCost	Readmit Accuracy Rate	Non-Readmit False
DT2_5xCost	DT2	5	1	903	59.92%	28.94%
DT3_5xCost	DT3	5	1	953	57.63%	30.47%
NN2_5xCost	NN2	5	1	1306	100.00%	100.00%
NN3_5xCost	NN3	5	1	1306	100.00%	100.00%
LR2_5xCost	LR2	5	1	1306	100.00%	100.00%
LR3_5xCost	LR3	5	1	1306	100.00%	100.00%
DT2_4xCost	DT2	4	1	773	50.38%	19.37%
DT3_4xCost	DT3	4	1	778	53.05%	21.90%
NN2_4xCost	NN2	4	1	1306	100.00%	100.00%
NN3_4xCost	NN3	4	1	1306	100.00%	100.00%
LR2_4xCost	LR2	4	1	1306	100.00%	100.00%
LR3_4xCost	LR3	4	1	1306	100.00%	100.00%
DT2_1.5xCost	DT2	3	2	707	12.60%	0.77%
DT3_1.5xCost	DT3	3	2	662	17.56%	0.54%
NN2_1.5xCost	NN2	3	2	1452	86.26%	51.45%
NN3_1.5xCost	NN3	3	2	1605	88.93%	58.12%
LR2_1.5xCost	LR2	3	2	2079	65.27%	69.14%
LR3_1.5xCost	LR3	3	2	2091	66.03%	69.83%

Questions?

Global Sponsor: 

Thank You for Attending

Global Sponsor: