Finding Rare Needles in the Haystack
Predictive Modeling with Unbalanced Data

Paul Bradley, *Chief Scientist, MethodCare, Inc.*
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…”
- 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

<table>
<thead>
<tr>
<th>Structure</th>
<th>WDBC-DT-Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area Mean</td>
<td>Microsoft_Decision_Trees</td>
</tr>
<tr>
<td>Area STD Err</td>
<td></td>
</tr>
<tr>
<td>Compactness Mean</td>
<td></td>
</tr>
<tr>
<td>Compactness STD Err</td>
<td></td>
</tr>
<tr>
<td>Concavity Mean</td>
<td></td>
</tr>
<tr>
<td>Concavity STD Err</td>
<td></td>
</tr>
<tr>
<td>Diagnosis</td>
<td>PredictOnly</td>
</tr>
<tr>
<td>Fractal Dimension Mean</td>
<td></td>
</tr>
<tr>
<td>Fractal Dimension STD Err</td>
<td></td>
</tr>
<tr>
<td>ID Number</td>
<td>Key</td>
</tr>
<tr>
<td>Perimeter Mean</td>
<td></td>
</tr>
<tr>
<td>Perimeter STD Err</td>
<td></td>
</tr>
<tr>
<td>Radius Mean</td>
<td></td>
</tr>
<tr>
<td>Radius STD Err</td>
<td></td>
</tr>
<tr>
<td>Smoothness Mean</td>
<td></td>
</tr>
<tr>
<td>Smoothness STD Err</td>
<td></td>
</tr>
<tr>
<td>Symmetry Mean</td>
<td></td>
</tr>
<tr>
<td>Symmetry STD Err</td>
<td></td>
</tr>
<tr>
<td>Texture Mean</td>
<td></td>
</tr>
<tr>
<td>Texture STD Err</td>
<td></td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Actual_Class</th>
<th>Predicted_Class</th>
<th>num_cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>357</td>
</tr>
<tr>
<td>M</td>
<td>B</td>
<td>26</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Default</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPLEXITY_PENALTY</td>
<td>1E-07</td>
<td></td>
<td>(0.0,1.0)</td>
</tr>
<tr>
<td>FORCE_REGRESSOR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAXIMUM_INPUT_ATTRIBUTES</td>
<td>0</td>
<td>255</td>
<td>[0,65535]</td>
</tr>
<tr>
<td>MAXIMUM_OUTPUT_ATTRIBUTES</td>
<td>255</td>
<td></td>
<td>[0,65535]</td>
</tr>
<tr>
<td>MINIMUM_SUPPORT</td>
<td>2</td>
<td>10.0</td>
<td>(0.0,...)</td>
</tr>
<tr>
<td>SCORE_METHOD</td>
<td>4</td>
<td></td>
<td>1,3,4</td>
</tr>
<tr>
<td>SPLIT_METHOD</td>
<td>3</td>
<td></td>
<td>[1,3]</td>
</tr>
</tbody>
</table>

Description:
Inhibits the growth of the decision tree. Decreasing the value increases the likelihood of a split, while increasing the value decreases the likelihood. The default value is based on the number of attributes for a given model: The default is 0.5 if there is 1 to 9 attributes; the default is 0.9 if there are 10 to 99 attributes; and the
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`

<table>
<thead>
<tr>
<th>Actual_Class</th>
<th>Predicted_Class</th>
<th>num_cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>356</td>
</tr>
<tr>
<td>B</td>
<td>M</td>
<td>1</td>
</tr>
<tr>
<td>M</td>
<td>B</td>
<td>5</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>21</td>
</tr>
</tbody>
</table>
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:

- 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:

- Example

```sql
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
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

  - 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)

```sql
SELECT MC1.TNumber,
       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 UBC_MetaCost1 Predictions MC1
INNER JOIN UBC_MetaCost2 Predictions MC2 ON MC1.TNumber = MC2.TNumber
INNER JOIN UBC_MetaCost3 Predictions MC3 ON MC1.TNumber = MC3.TNumber
INNER JOIN UBC_MetaCost4 Predictions MC4 ON MC1.TNumber = MC4.TNumber
INNER JOIN UBC_MetaCost5 Predictions MC5 ON MC1.TNumber = MC5.TNumber
```
Average the Probabilities

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

<table>
<thead>
<tr>
<th>IDNumber</th>
<th>Prob_Malignant_1</th>
<th>Prob_Malignant_2</th>
<th>Prob_Malignant_3</th>
<th>Prob_Malignant_4</th>
<th>Prob_Malignant_5</th>
<th>AvgProb_Malignant</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.05326686002522068</td>
<td>0.00780864595967286</td>
<td>0.0152838427947557</td>
<td>0.0100502512562612</td>
<td>0.00738084596677285</td>
<td>0.0201129772495167</td>
</tr>
<tr>
<td>2</td>
<td>0.05326686002522068</td>
<td>0.0239254710327457</td>
<td>0.064046163922526</td>
<td>0.0100502512562612</td>
<td>0.00738084596677285</td>
<td>0.0560936891075247</td>
</tr>
<tr>
<td>3</td>
<td>0.05326686002522068</td>
<td>0.0398006459967286</td>
<td>0.3152838427947557</td>
<td>0.0100502512562612</td>
<td>0.00738084596677285</td>
<td>0.0201129772495167</td>
</tr>
<tr>
<td>4</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>5</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>6</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>7</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>8</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>9</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>10</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>11</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>12</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>13</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>14</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>15</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>16</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>17</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
<tr>
<td>18</td>
<td>0.05326686002522068</td>
<td>0.33672131147541</td>
<td>0.445255474452555</td>
<td>0.010046372239576</td>
<td>0.104591394148021</td>
<td>0.30830981473962</td>
</tr>
</tbody>
</table>
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’:
    - \((\text{Avg Prob(Benign)}) \times (\text{Cost of predicting ‘Benign’ when actually ‘Malignant’})\)
  - Risk of predicting ‘Malignant’ when actually ‘Benign’:
    - \((\text{Avg Prob(Malignant)}) \times (\text{Cost of predicting ‘Malignant’ when actually ‘Benign’})\)
  - Choose the class label the minimizes risk:
    - If \((\text{Risk of predicting ‘Benign’ when actually ‘Malignant’}) < (\text{Risk of predicting ‘Malignant’ when actually ‘Benign’})\), then label = ‘Benign’
    - Else label = ‘Malignant’
Updated Class Labels – Example

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

<table>
<thead>
<tr>
<th>Actual_Class</th>
<th>Predicted_Class</th>
<th>num_cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B</td>
<td>302</td>
</tr>
<tr>
<td>B</td>
<td>M</td>
<td>55</td>
</tr>
<tr>
<td>M</td>
<td>B</td>
<td>7</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>19</td>
</tr>
</tbody>
</table>
## Example MetaCost for Readmissions

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