Implementing Instant-Book and Improving Customer Service Satisfaction

Arturo Heyner Cano Bejar, Nick Danks Kellan Nguyen, Tonny Kuo
Problem statement

**Problem:** high rejection rate (15%) → lost sales and customer dissatisfaction

**Strategy:** Provide a tool for AsiaYo! to pipeline transactions according to risk of rejection.

**Goal:** Implement Instant-Booking service

Stakeholder

- AsiaYo Management Team
- AsiaYo Customer Service Team
- Guest / Host
- Competitors: Airbnb, booking.com, Agoda

Opport. / Challenge

- Increase *revenues*
  - Company will employ *service team* more efficiently.
- Manage AsiaYo! resources more efficiently.
- Higher *customer satisfaction*
**Data Mining Goal**

<table>
<thead>
<tr>
<th>Predicting</th>
<th>Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicting the probability of a transaction being rejected by the host.</td>
<td>Clustering similar facilities to offer alternatives to guests.</td>
</tr>
<tr>
<td>Low risk: instant-book</td>
<td></td>
</tr>
<tr>
<td>High risk: Service team</td>
<td></td>
</tr>
</tbody>
</table>

**Outcome Variable**

1. Probability of rejection (%)
2. Binary for Rejection (cut-off value very important)
3. Clusters of similar facilities

**Methods**

**Classification**
- KNN
- Classification Trees
- Logistic regression
- Neural Networks
- Naive Bayes
- Ensembles
- Discriminant Analysis

**Clustering**
- Clustering analysis
Data

- **Data source and size:** AsiaYo!, 65534 observations
- **Unit of Analysis:** One booking transaction
- **Output:** new.ack.status
- **Input variables:**
  - **Predicting:** Guests, nights, rooms, amount_paid, DOW.ci, DOW.created.at, advancebook, loc_popularity, nationality
  - **Clustering:** Accom_fac, room_fac, room_bath_fac, location
Methods & Performance Evaluation

- Task: Classification (supervised); Clustering (unsupervised)
- Benchmark: naive (the most popular class)
- Relevant performance measures:
  - False Positive (very important -> lost sales / increased effort by the team)
  - False Negative (host dissatisfaction, inactive host)
- Relevance to business problem
  - Sales
  - Customer satisfaction
Implementation & Production Considerations

**Implementation**
- IT team
- Service team: Manage high risk bookings
- R&D team: Develop different intervention

**Production Consideration**
- Real time (At booking).
- One-time analysis
- The model should be re-analyzed weekly and sensitivity and specificity reviewed and costs re-evaluated
I update our new slides as belows

- Cover: Informative title, team number and member names
- Business problem (stakeholder, challenge/opportunity, humanity considerations)
- Data mining problem (supervised/unsupervised, explanatory/predictive, how to be deployed)
- Data description (what is a row? Output and input variables; partitioning)
- Methods (methods, relevant outputs)
- Evaluation (metrics of interest, benchmark, comparison)
- Recommendations
Identifying high-risk rejection orders to improve customer service satisfaction

Arturo Heyner Cano Bejar
Nick Danks
Kellan Nguyen

Presenter: Tonny Meng-Lun Kuo
Advisor: Prof. Galit Shmueli
## Business Problem

### Problem statement

**Problem**: high rejection rate (15%) → lost sales and customer dissatisfaction

**Strategy**: Provide a tool for AsiaYo! to identify high-risk rejection orders.

**Goal**: Rank transactions with high rejection prob. of rate.

### Stakeholder

- AsiaYo Management Team
- AsiaYo Customer Service Team
- Guest / Host
- Competitors: Airbnb, booking.com, Agoda

### Opport. / Challenge

Increase **revenues** by less rejections and faster intervention

Company will employ **service team** more **efficiently**.

Manage AsiaYo! resources more efficiently.

Higher **customer satisfaction**
# Data Mining Goal

**Ranking** the probability of a transaction being rejected by the host. *(supervised goal)*

- Low risk: Normal intervention
- High risk: Direct intervention from Service team

**Outcome Variable**

Binary for Rejection (cut-off value = 0.5)

**Methods**

- Classification
  - Logistics Regression
  - KNN
  - Naive Bayes
  - Discriminant Analysis
  - SVM
  - Classification Tree
  - Boosted Trees
  - Random Forest

**Unbalanced data**

- Over-sampling
• **Data source and size:** AsiaYo!, 59265 observations
• **Unit of Analysis:** One booking transaction
• **Output:** is.rejected [derived from new.ack.status]
• **Input variables:**
  ○ Numeric: guests, nights, rooms, amount_paid, advancebook
  ○ Factor: DOW.ci, DOW.created.at
• **Data partitions:**
  ○ Training (40%), Validation (30%), Test (30%)
Methods & Performance Evaluation

- Task: Ranking (supervised)
- Benchmark: naive (the most popular class)
- Relevant performance measures:
  - Sensitivity
  - Lift Chart
  - False Positive (Important -> lost sales / increased effort of team)
  - False Negative (host dissatisfaction, inactive host)
# Empirical Results (Descriptive data)

## Method: Non-oversampling

<table>
<thead>
<tr>
<th></th>
<th># Records</th>
<th>% is.rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training (40%)</td>
<td>29,632</td>
<td>15.13%</td>
</tr>
<tr>
<td>Validation (30%)</td>
<td>17,779</td>
<td>14.69%</td>
</tr>
<tr>
<td>Testing (30%)</td>
<td>11,854</td>
<td>15.50%</td>
</tr>
</tbody>
</table>

## Method: Oversampling

<table>
<thead>
<tr>
<th></th>
<th># Records</th>
<th>% is.rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training (40%)</td>
<td>7,182</td>
<td>50%</td>
</tr>
<tr>
<td>Validation (30%)</td>
<td>17,779</td>
<td>15.25%</td>
</tr>
<tr>
<td>Testing (30%)</td>
<td>17,780</td>
<td>14.78%</td>
</tr>
</tbody>
</table>

## Non-oversampling

<table>
<thead>
<tr>
<th>Methods</th>
<th>Sensitivity</th>
<th>Accuracy</th>
<th>Specificity</th>
<th>TP</th>
<th>FP</th>
<th>TN</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistics</td>
<td>0.00</td>
<td>0.85</td>
<td>1.00</td>
<td>4</td>
<td>1833</td>
<td>10013</td>
<td>4</td>
</tr>
<tr>
<td>KNN</td>
<td>0.00</td>
<td>0.85</td>
<td>1.00</td>
<td>0</td>
<td>1837</td>
<td>10017</td>
<td>0</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.03</td>
<td>0.84</td>
<td>0.99</td>
<td>60</td>
<td>1777</td>
<td>9899</td>
<td>118</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>0.00</td>
<td>0.84</td>
<td>1.00</td>
<td>2</td>
<td>1835</td>
<td>10013</td>
<td>4</td>
</tr>
<tr>
<td>SVM</td>
<td>0.00</td>
<td>0.85</td>
<td>1.00</td>
<td>0</td>
<td>1837</td>
<td>10017</td>
<td>0</td>
</tr>
<tr>
<td>Classification Tree</td>
<td>0.01</td>
<td>0.83</td>
<td>0.98</td>
<td>15</td>
<td>1822</td>
<td>9865</td>
<td>152</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.13</td>
<td>0.84</td>
<td>0.97</td>
<td>231</td>
<td>1606</td>
<td>9711</td>
<td>306</td>
</tr>
<tr>
<td>Boosted Tree</td>
<td>0.20</td>
<td>0.81</td>
<td>0.92</td>
<td>364</td>
<td>1473</td>
<td>9227</td>
<td>790</td>
</tr>
</tbody>
</table>

## Oversampling

<table>
<thead>
<tr>
<th>Methods</th>
<th>Sensitivity</th>
<th>Accuracy</th>
<th>Specificity</th>
<th>TP</th>
<th>FP</th>
<th>TN</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistics</td>
<td>0.65</td>
<td>0.55</td>
<td>0.53</td>
<td>1705</td>
<td>923</td>
<td>8023</td>
<td>7129</td>
</tr>
<tr>
<td>KNN</td>
<td>0.00</td>
<td>0.85</td>
<td>1.00</td>
<td>0</td>
<td>2628</td>
<td>15152</td>
<td>0</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.69</td>
<td>0.53</td>
<td>0.51</td>
<td>1801</td>
<td>827</td>
<td>7694</td>
<td>7458</td>
</tr>
<tr>
<td>Discriminant Analysis</td>
<td>0.65</td>
<td>0.52</td>
<td>0.50</td>
<td>1721</td>
<td>907</td>
<td>7603</td>
<td>7549</td>
</tr>
<tr>
<td>SVM</td>
<td>0.58</td>
<td>0.63</td>
<td>0.64</td>
<td>1530</td>
<td>1098</td>
<td>9633</td>
<td>5519</td>
</tr>
<tr>
<td>Classification Tree</td>
<td>0.39</td>
<td>0.57</td>
<td>0.60</td>
<td>1034</td>
<td>1594</td>
<td>9084</td>
<td>6068</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.63</td>
<td>0.62</td>
<td>0.61</td>
<td>1664</td>
<td>964</td>
<td>9317</td>
<td>5835</td>
</tr>
<tr>
<td>Boosted Tree</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
<td>1581</td>
<td>1047</td>
<td>9096</td>
<td>6056</td>
</tr>
</tbody>
</table>
Empirical Results (Non-oversampling)

- In non-oversampling, the performance of overall accuracy are similar (around 80%).
- **Boosted tree** and **random forest** are top two methods in sensitivity.
Empirical Results (Oversampling)

- In oversampling, KNN gets the highest accuracy but lowest sensitivity.
- **Naive Bayes** and **discriminant analysis** get better performance in sensitivity.
1. This project identifies **transactions with higher probability** to be rejected using data mining algorithms to reduce dissatisfaction and increase profits.

2. Due to the **unbalanced dataset** and **ranking goal**, we suggest to adopt **oversampling with Naive Bayes method** to build the predictive model.

3. Although we can make the prediction based on the current datasets (accuracy = 0.69), **more derived variables** could be collected and included in predictive model for performance improvement.
   - dynamic popularity: popularity of the properties at specific time
   - property location: location of the properties
   - host’s commitment: the degree of how hosts’ commitment to the platform
   - Seasonal popularity: whether its a national/international holidays