Using a Random Forest model to predict enrollment

Ward Headstrom
Institutional Research
Humboldt State University
CAIR 2013
Overview

- Forecasting enrollment to assist University planning
- The R language
- The Random Forest model
- Binary Logistic Regression model
- Cautions and Conclusions

- The example I am going to use is projecting **New** enrollment. These techniques can easily be applied to predicting...
  - Retention
  - Graduation
  - Other future events
Simple enrollment projections

1) how many students enrolled last year?
2) enhance by breaking it down into subgroups
3) possibly use linear regressions (trends)

<table>
<thead>
<tr>
<th>Applicant Counts Report Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time frame</td>
</tr>
<tr>
<td>Semester</td>
</tr>
<tr>
<td>Cohort</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Final Fall Registered</th>
</tr>
</thead>
<tbody>
<tr>
<td>report generated: 03-OCT-13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Applicant type</th>
<th>Fall 2007</th>
<th>Fall 2008</th>
<th>Fall 2009</th>
<th>Fall 2010</th>
<th>Fall 2011</th>
<th>Fall 2012</th>
<th>Fall 2013</th>
<th>Fall 2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>First-time UG</td>
<td>1,058</td>
<td>1,202</td>
<td>1,382</td>
<td>1,316</td>
<td>1,292</td>
<td>1,242</td>
<td>1,369</td>
<td></td>
</tr>
<tr>
<td>Lower-div xfer</td>
<td>302</td>
<td>156</td>
<td>198</td>
<td>103</td>
<td>149</td>
<td>141</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Upper-div xfer</td>
<td>636</td>
<td>612</td>
<td>572</td>
<td>786</td>
<td>795</td>
<td>807</td>
<td>921</td>
<td></td>
</tr>
<tr>
<td>Returning UG</td>
<td>110</td>
<td>102</td>
<td>99</td>
<td>84</td>
<td>108</td>
<td>108</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>Masters</td>
<td>199</td>
<td>185</td>
<td>174</td>
<td>175</td>
<td>133</td>
<td>147</td>
<td>182</td>
<td></td>
</tr>
<tr>
<td>Credential</td>
<td>168</td>
<td>117</td>
<td>124</td>
<td>118</td>
<td>111</td>
<td>107</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>Second Bachelor</td>
<td>63</td>
<td>47</td>
<td>35</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Unclassified PB</td>
<td>17</td>
<td>7</td>
<td>10</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Transitory</td>
<td>176</td>
<td>183</td>
<td>30</td>
<td>71</td>
<td>99</td>
<td>43</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>2,729</td>
<td>2,611</td>
<td>2,624</td>
<td>2,660</td>
<td>2,693</td>
<td>2,599</td>
<td>2,749</td>
<td></td>
</tr>
</tbody>
</table>

4) enhance further by looking at to-date information
2014 projection = 2013 “to-date” yield * 2014 apps = \( \frac{1,369}{3,692} \times 4,361 = 1,617 \)
<table>
<thead>
<tr>
<th>Student type</th>
<th>Total Headcount</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fall 2013</td>
<td>Fall 2014</td>
<td>%change</td>
<td>Fall 2013</td>
<td>Fall 2014</td>
<td>%change</td>
<td></td>
</tr>
<tr>
<td>Continuing Undergrad</td>
<td>5,299</td>
<td>5,624</td>
<td>5.8%</td>
<td>4,874</td>
<td>5,173</td>
<td>5.8%</td>
<td></td>
</tr>
<tr>
<td>Returning Undergrad</td>
<td>96</td>
<td>96</td>
<td>0.0%</td>
<td>75</td>
<td>75</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>First-time Undergrad</td>
<td>1,368</td>
<td>1,617</td>
<td>15.4%</td>
<td>1,277</td>
<td>1,509</td>
<td>15.4%</td>
<td></td>
</tr>
<tr>
<td>Transfer Undergrad</td>
<td>971</td>
<td>930</td>
<td>-4.4%</td>
<td>875</td>
<td>838</td>
<td>-4.4%</td>
<td></td>
</tr>
<tr>
<td>Continuing/Returning Postbac</td>
<td>249</td>
<td>256</td>
<td>2.6%</td>
<td>172</td>
<td>177</td>
<td>2.6%</td>
<td></td>
</tr>
<tr>
<td>New Postbac</td>
<td>278</td>
<td>283</td>
<td>1.8%</td>
<td>290</td>
<td>295</td>
<td>1.8%</td>
<td></td>
</tr>
<tr>
<td>Transitory</td>
<td>32</td>
<td>32</td>
<td>0.0%</td>
<td>20</td>
<td>20</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>8,293</td>
<td>8,838</td>
<td>6.2%</td>
<td>7,583</td>
<td>8,088</td>
<td>6.2%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student type</th>
<th>Total Headcount</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Spring 14</td>
<td>Spring 15</td>
<td>%change</td>
<td>Spring 14</td>
<td>Spring 15</td>
<td>%change</td>
<td></td>
</tr>
<tr>
<td>Continuing Undergrad</td>
<td>6,976</td>
<td>7,457</td>
<td>6.5%</td>
<td>6,348</td>
<td>6,786</td>
<td>6.5%</td>
<td></td>
</tr>
<tr>
<td>Returning Undergrad</td>
<td>21</td>
<td>21</td>
<td>0.0%</td>
<td>19</td>
<td>19</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>First-time Undergrad</td>
<td>39</td>
<td>39</td>
<td>0.0%</td>
<td>35</td>
<td>35</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>Transfer Undergrad</td>
<td>408</td>
<td>408</td>
<td>0.0%</td>
<td>371</td>
<td>371</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>Continuing/Returning Postbac</td>
<td>430</td>
<td>439</td>
<td>2.1%</td>
<td>391</td>
<td>400</td>
<td>2.1%</td>
<td></td>
</tr>
<tr>
<td>New Postbac</td>
<td>29</td>
<td>29</td>
<td>0.0%</td>
<td>26</td>
<td>26</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>Transitory</td>
<td>39</td>
<td>39</td>
<td>0.0%</td>
<td>35</td>
<td>35</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>Totals</td>
<td>7,941</td>
<td>8,432</td>
<td>5.8%</td>
<td>7,226</td>
<td>7,673</td>
<td>5.8%</td>
<td></td>
</tr>
</tbody>
</table>
### Humboldt Institutional Summaries

#### Applicant Counts

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First-time UG</td>
<td>107</td>
<td>94</td>
<td>96</td>
<td>6</td>
<td>107</td>
<td>97</td>
<td>12</td>
<td>117</td>
</tr>
<tr>
<td>Lower-div xfer</td>
<td>157</td>
<td>212</td>
<td>122</td>
<td>5</td>
<td>77</td>
<td>146</td>
<td>29</td>
<td>54</td>
</tr>
<tr>
<td>Upper-div xfer</td>
<td>424</td>
<td>513</td>
<td>615</td>
<td>115</td>
<td>677</td>
<td>697</td>
<td>277</td>
<td>749</td>
</tr>
<tr>
<td>Returning UG</td>
<td>52</td>
<td>57</td>
<td>58</td>
<td>23</td>
<td>107</td>
<td>120</td>
<td>43</td>
<td>133</td>
</tr>
<tr>
<td>Masters</td>
<td>45</td>
<td>58</td>
<td>78</td>
<td>12</td>
<td>36</td>
<td>57</td>
<td>53</td>
<td>84</td>
</tr>
<tr>
<td>Credential</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Second Bachelor</td>
<td>55</td>
<td>47</td>
<td>72</td>
<td>22</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Unclassified PB</td>
<td>16</td>
<td>18</td>
<td>13</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transitory</td>
<td>26</td>
<td>41</td>
<td>45</td>
<td>23</td>
<td>10</td>
<td>1</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

**Totals**: 885 - 1,044 - 1,100 - 214 - 1,024 - 1,126 - 419 - 1,147

---

Return to Institutional Summary Index
But what about…

• Why applicant yield might not be the best predictor:
  • Admits more likely to enroll
  • Confirms more likely to enroll
  • Denied or withdrawn will not enroll
  • Housing deposits may be good indicator of intent
  • Local applicants more likely than distant applicants
  • Certain majors may be more likely to enroll
  • White applicants may be more likely than URM

• Ideally, we would like to use all the data we have about applicants to predict how likely they are to enroll.
  • Variables: demographics, academics, actions to-date
  • Model 1: Random Forest
  • Model 2: Binary logistic regression
Data files

• All the data fields you think might help predict yields
  • Major discipline
  • Region of origin
  • Sex
  • Ethnicity
  • Academic preparation

• Actions
  • Visited campus
  • Confirmed intent to enroll
  • Paid housing deposit

• Be careful of institutional actions
  • admission
  • cancellation
The language R

CAIR comment: an emphasis on R would be “limiting to institutions that used other software”.

• The first (only?) implementation of Random Forest models
• R is open source – free to use
  • [http://cran.us.r-project.org/](http://cran.us.r-project.org/)
• Many online tutorials:
  • [http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf](http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf)
  • [https://www.coursera.org/course/compdata](https://www.coursera.org/course/compdata)
• [www.researchgate.net/post/Which_is_better_R_or_SPSS](http://www.researchgate.net/post/Which_is_better_R_or_SPSS)
R and RStudio overview

- 4 panes – help, history, import dataset, packages
- Function-based: function(data,options)
- Case-specific language
- Object types: data.frame, vector, scalars, factor, models
- Useful commands:
  - command line as calculator
  - assignment -> or <-
  - functions: na.omit(), summary(), table(), tolower()
  - subsets: dataframe[row select, column select]
  - graphics: hist(), plot()
  - library(), especially library(randomForest)
RStudio

```r
# Load the dataset
train <- apps_td_spring[train$termcode == 2112,]

test1 <- apps_td_spring[train$termcode == 2132,]

rf <- randomForest(formula = centrd ~ class + apptypelet + status + acceptance + housingdep + intent, data = train)

Call:
randomForest(formula = centrd ~ class + apptypelet + status + acceptance + housingdep + intent, data = train)
Type of random forest: classification
Number of trees: 500

No. of variables tried at each split: 4

OOB estimate of error rate: 12.31%

confusion matrix:
          N    Y class.error
N 1415 146 0.08332979
Y 145 658 0.1807275

varImpPlot(rf)
```
Import data into R

> apps_td_spring <- read.csv("C:/Users/ward/Google Drive/IRP/CAIR 2013/apps_td_spring.csv")
> summary(apps_td_spring)
Decision Trees

```r
> tr <- tree(CENREG ~ ACCEPTSUG_T + INTENT_T + STATUS, data = apps_td_spring)
> plot(tr)
> text(tr)
```
Random Forest Model

• Developed by Leo Brieman and Adele Cutler
• Plan: grow a random forest of 500 decision trees
  • randomForest(cenreg~variable1+variable2+...,data=train)
  • Randomly picks fields for each tree
  • Randomly selects rows to exclude from each tree
• Measure of variable importance
• Out Of Box estimate of error rate and Confusion matrix

```
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 4

OOB estimate of error rate: 12.52%
Confusion matrix:
    N     Y  class.error
 N 1414  147  0.0941704
 Y  149  654  0.1855542
```

• Run new data through all 500 trees and let them vote
Random Forest model of applicant yield

```r
> names(apps_td_spring) <- tolower(names(apps_td_spring))
> table(apps_td_spring$termcode)

2102 2112 2122 2132 2142
 214 1024 1126  419  1147

> train <- apps_td_spring[apps_td_spring$termcode < 2132,]
> test <- apps_td_spring[apps_td_spring$termcode == 2132,]
> project <- apps_td_spring[apps_td_spring$termcode > 2132,]
> rf <- randomForest(cenreg ~ class + apptypelet + status + ethnicity + urm,
> rf

Call:
  randomForest(formula = cenreg ~ class + apptypelet + status +
acceptsug_td + housingdep_td + intent_td, data = train)
  Type of random forest: classification
  No. of variables tried at each split: 4

OOB estimate of  error rate: 12.31%

Confusion matrix:
   N Y class.error
  N 1415 146  0.09352979
  Y 145 658  0.18057285

> varImpPlot(rf)
```
varImpPlot(rf)
For categorical predictors, the splitting point is represented by an integer, whose binary expansion gives the identities of the categories that go to left or right. For example, if a predictor has four categories, and the split point is 13. The binary expansion of 13 is (1, 0, 1, 1) (because $13 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3$), so cases with categories 1, 3, or 4 in this predictor get sent to the left, and the rest to the right.
Testing and making a Projection

Random Forest projects that 42% of current Spring apps will enroll, compared to 45% of last year’s apps to-date and 34% of training years’.
Binary Logistic Regression

- \( p(x) \) is the probability that \( x \) will occur, where \( x \) is a binary object (Y/N, 1/0, true/false)
- \( \log \left( \frac{p(x)}{1-p(x)} \right) = B_0 + B_1 \times X_1 + B_2 \times X_2 + B_3 \times X_3 + \ldots \)
- \( B_n \) represents calculated coefficients
- \( X_n \) represents the value of dependent variables
- Break up factor variables into many terms where \( X_n \) is 1 or 0
- Can manipulate the result to return the probability (between 0 and 1) that \( x \) will occur, given the state of a particular set of dependent variables.
- Difficult to predict outcome of a single individual
- Can sum probabilities to estimate total
Binary logistic regression model of applicant yield

```r
> blr <- glm(cenreg ~ status + acceptsug_t + intent_t, data = train, family = binomial)
> summary(blr)

Call:
  glm(formula = cenreg ~ status + acceptsug_t + intent_t, family = binomial,
     data = train)

Deviance Residuals:
     Min       1Q   Median       3Q      Max
-2.8129  -0.5208  -0.4253  0.1125  2.6771

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                -2.3575     0.1225  -19.246  < 2e-16 ***
status04-Withdrew app      -0.4305     0.7284  -0.591   0.5545
status05-Complete          1.0736     0.2348   4.573  4.80e-06 ***
status05-In Review         0.4282     0.3183   1.345   0.1785
status06-Denied           -1.1979     0.5217  -2.296   0.0217 *
status07-Admitted          0.8117     0.1709   4.748   2.05e-06 ***
status08-Not coming       -14.2086   438.0949  -0.032   0.9741
status10-Housing deposit   0.5734     0.8411   0.682   0.4955
status10-Intends to enroll 1.1399     0.2370   4.809  1.52e-06 ***
acceptsug_t               6.2944     1.0096   6.234  4.54e-10 ***
intent_t                  2.3712     0.1984  11.954  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3029.8  on 2363  degrees of freedom
Residual deviance: 1643.7  on 2353  degrees of freedom
AIC: 1665.7

Number of Fisher Scoring iterations: 15
```
> anova(blr,test="chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: cenreg

Terms added sequentially (first to last)

<table>
<thead>
<tr>
<th>Term</th>
<th>Df</th>
<th>Deviance</th>
<th>Resid. Df</th>
<th>Resid. Dev</th>
<th>Pr(&gt;Chi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NULL</td>
<td></td>
<td></td>
<td>2363</td>
<td>3029.8</td>
<td></td>
</tr>
<tr>
<td>class</td>
<td>8</td>
<td>80.07</td>
<td>2355</td>
<td>2949.7</td>
<td>4.743e-14***</td>
</tr>
<tr>
<td>apptypelet</td>
<td>5</td>
<td>30.78</td>
<td>2350</td>
<td>2919.0</td>
<td>1.035e-05***</td>
</tr>
<tr>
<td>status</td>
<td>8</td>
<td>822.38</td>
<td>2342</td>
<td>2096.6</td>
<td>&lt; 2.2e-16***</td>
</tr>
<tr>
<td>ethnicity</td>
<td>7</td>
<td>29.78</td>
<td>2335</td>
<td>2066.8</td>
<td>0.000104***</td>
</tr>
<tr>
<td>urm</td>
<td>2</td>
<td>5.08</td>
<td>2333</td>
<td>2061.7</td>
<td>0.078694.</td>
</tr>
<tr>
<td>sex</td>
<td>2</td>
<td>2.29</td>
<td>2331</td>
<td>2059.4</td>
<td>0.317455</td>
</tr>
<tr>
<td>origin_region</td>
<td>11</td>
<td>38.74</td>
<td>2320</td>
<td>2020.7</td>
<td>5.863e-05***</td>
</tr>
<tr>
<td>residency</td>
<td>6</td>
<td>43.73</td>
<td>2314</td>
<td>1977.0</td>
<td>8.357e-08***</td>
</tr>
<tr>
<td>low_income</td>
<td>1</td>
<td>42.23</td>
<td>2313</td>
<td>1934.7</td>
<td>8.131e-11***</td>
</tr>
<tr>
<td>firstgen</td>
<td>2</td>
<td>12.23</td>
<td>2311</td>
<td>1922.5</td>
<td>0.002211**</td>
</tr>
<tr>
<td>admit</td>
<td>0</td>
<td>0.00</td>
<td>2311</td>
<td>1922.5</td>
<td></td>
</tr>
<tr>
<td>isource</td>
<td>8</td>
<td>23.25</td>
<td>2303</td>
<td>1899.2</td>
<td>0.003060**</td>
</tr>
<tr>
<td>appfeestat</td>
<td>9</td>
<td>27.71</td>
<td>2294</td>
<td>1871.5</td>
<td>0.001066**</td>
</tr>
<tr>
<td>discipline</td>
<td>21</td>
<td>37.19</td>
<td>2273</td>
<td>1834.3</td>
<td>0.015998*</td>
</tr>
<tr>
<td>acceptsug_td</td>
<td>1</td>
<td>303.95</td>
<td>2272</td>
<td>1530.4</td>
<td>&lt; 2.2e-16***</td>
</tr>
<tr>
<td>housingdep_td</td>
<td>1</td>
<td>65.38</td>
<td>2271</td>
<td>1465.0</td>
<td>6.176e-16***</td>
</tr>
<tr>
<td>intent_td</td>
<td>1</td>
<td>157.64</td>
<td>2270</td>
<td>1307.4</td>
<td>&lt; 2.2e-16***</td>
</tr>
</tbody>
</table>

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```r
> train$p <- predict(blr, train, type = "response")

> table(round(train$p, 1), train$cenreg)

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>729</td>
<td>13</td>
</tr>
<tr>
<td>0.1</td>
<td>501</td>
<td>43</td>
</tr>
<tr>
<td>0.2</td>
<td>128</td>
<td>45</td>
</tr>
<tr>
<td>0.3</td>
<td>55</td>
<td>19</td>
</tr>
<tr>
<td>0.4</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>0.5</td>
<td>18</td>
<td>29</td>
</tr>
<tr>
<td>0.6</td>
<td>26</td>
<td>44</td>
</tr>
<tr>
<td>0.7</td>
<td>32</td>
<td>71</td>
</tr>
<tr>
<td>0.8</td>
<td>27</td>
<td>112</td>
</tr>
<tr>
<td>0.9</td>
<td>19</td>
<td>79</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>324</td>
</tr>
</tbody>
</table>

> hist(train$p)
```
BLR model – testing and projecting

> sum(test$cenreg=="Y")
[1] 190
> test$p<-predict(blr,test,type="response")
> sum(test$p)
[1] 182.7157

> project$p<-predict(blr,project,type="response")
> sum(project$p)
[1] 324.4795

Binary Logistic Regression predicted 324 of current Spring applicants will enroll, compared to 340 projected by Random Forest model.
Cautions and Conclusions

• Null or new values in variables will cause problems

• Beware of to-date variables (e.g. intent_td). Make sure that procedures have not changed in a way that will affect behavior.

• R is a very powerful tool which can be very useful if you are willing to invest some time learning it.

• Multivariate models *may* improve the accuracy of your predictions. Corroborate with simple models and consultation with involved staff.
Questions? Comments?

This presentation:
www.humboldt.edu\irp\presentations\randomforest.pdf

My email:
Ward.Headstrom@humboldt.edu