Data Science Capstone - Erin Williams

October 3, 2017

One of the primary questions any educational institution must constantly ask is: Are students learning? Businesses, on the other hand, must ask the question: Are we profitable? As a non-profit school that relies on tuition revenue to pay its instructors and places a high value on student success, analyzing student enrollment, particularly dropout and repeat rates, will help us intelligently answer these two questions.

Turing School of Software and Design is a 7 month, intensive computer science program that turns beginners into competent developers. This entails four sessions of 60-80 hour weeks, six weeks at a time. Needless to say, it is not for the faint of heart, or those unwilling to work hard.

As a result, in order to ensure we have a strong student body full of people committed to complete the program, we have a rigorous admissions process, including a 5 step application and a 60-90 minute interview. Both the application and the interview include questions designed to determine grit, growth mindset, and logical aptitude.

Despite this, only 75% of students graduate within four modules, and 12% of students who enroll drop out before they complete the program. As an institution, we have a goal to reduce the rate of dropouts to 10% overall, and reduce the rate further, to less than 5% for students who leave after mod 1. Some churn will inevitably be due to circumstances beyond our control, such as health or family issues, but some students who leave do so for reasons that we can either anticipate or mitigate. I chose that subgroup (“dropouts”) to be the primary focus of my report.

In addition, our program allows students to repeat one or modules, distinguishing us from similar programs. We have set a goal that on average, 10-15% of students repeat one or more modules. These students typically repeat because they were unprepared or struggle in some way during the module. Building a model that would allow us to predict which students may need to repeat would help us offer support earlier than we currently do and perhaps prevent them from needing to repeat at all. “Repeaters” will be the secondary focus of this report.

Finally, the first filter for students before they enter the school is the application and interview process. We must have a strong, data-driven interview system in place in order to ensure that we are only enrolling those students who have a strong chance of

success. Therefore, I will also be analyzing a few select metrics from the interview process.

This data set can be categorized into the following groups:

1. Demographic information
    a. Age
    b. Race/ethnicity
    c. Gender
    d. Salary
        i. At last job prior to enrolling
    e. Education
        i. Level before enrolling
    f. Job
        i. Last job prior to enrolling
    g. Location
    h. Vet status
    i. Behavior in program
2. Enrollment information
    a. Cohort
    b. Enrollments
    c. Mod dropped out
    d. Mod repeated
    e. Times repeated
    f. Program
    g. Reason for withdrawal
    h. Status
3. Application/interview information
    a. Experience
        i. How much programming experience they had prior to joining the program
    b. Payment plan
    c. Referred by
    d. Quiz
        i. Application quiz score (out of 8)
    e. Interviewer
    f. Int logic score
        i. Score given by interviewer on collaborative logic portion
4. Identifiers, used as keys
    a. Id

b. App id

For the purposes of this data analysis, multiple data points must be compared in order to determine any patterns and create a prediction model.

Some data we don't have and cannot get includes: students who were not accepted who may have been successful, reasons students dropped out before beginning the program, and granular data on dropouts (exact timing, individual details). In addition, there are some columns of incomplete data because we only recently started collecting demographic information from enrolled students. Those categories which will mature as more data is collected are race/ethnicity, salary, job, education, experience, and age.

This data was drawn from multiple datasets from two different web applications, google surveys, and personal interviews. Wrangling the data involved merging those data sets, deleting redundant columns, and filling in missing data, both manually in excel and using R.

I have created some simple dot plots at this point that illustrate a predictable correlation between logic scores and interview logic scores (showing that our metrics are aligned). Just viewing the data it appears that students who are over 45 and students who are under 20 tend to struggle and have significantly higher dropout rates than the general population, as an example.

Based on my initial observations, the following are the business questions I would like to answer with this project:

1. Students with which demographic characteristics are most likely to drop out? (race, gender, logic scores, payment plan)
2. Are there factors which make a student more likely to repeat?
3. Can we calculate a student's likelihood of repeating and support them accordingly?
4. What are the "success rates" of different interviewers?
5. Are students with higher logic scores more likely to be successful completing the program?

Question 1 speaks to the primary dropout focus cited earlier in this paper. Questions 2 and 3 are related to repeaters. Questions 4 and 5 allow us to take a look at the interview process.

Machine learning approach

I chose to use logistic regression for this particular data set, with the independent variable being the binary column ‘graduated’, and various variables serving as the dependent variables.

The first question I asked was a combination of questions 1 and 5 above: *What is the likelihood of graduation for students from various demographic groups based on their initial logic score?*

I started with the following glm call:

```
log.reg.all <-
glm(graduated~age+quiz+GenderBinary+AfAm+Asian+Indian+Latinx+White+MENA+ve
terans+BE+FE+enrollments+int_logic_score, data=alldata_safe1, family="binomial")

Call:  glm(formula = graduated ~ age + quiz + GenderBinary + AfAm +
           Asian + Indian + Latinx + White + MENA + veterans + BE +
           FE + enrollments + int_logic_score, family = "binomial",
         data = alldata_safe1)

Coefficients:
  (Intercept)           age          quiz  GenderBinary          AfAm         Asian
Indian       Latinx       White        MENA    veterans          BE          FE
-8.59629     -0.02051      0.40771      0.20075      0.12910      0.30368
0.53739      1.31899      0.97888           NA     -0.21321     -0.43124
NA
enrollments  int_logic_score
1.32030      0.16522

Degrees of Freedom: 436 Total (i.e. Null);  424 Residual
(157 observations deleted due to missingness)
Null Deviance:          530.3
Residual Deviance: 341.1  AIC: 367.1

summary(log.reg.all)
Call:
  glm(formula = graduated ~ age + quiz + GenderBinary + AfAm +
      Asian + Indian + Latinx + White + MENA + veterans + BE +
```

```
    FE + enrollments + int_logic_score, family = "binomial",
    data = alldata_safe1)

Deviance Residuals:
  Min       1Q   Median       3Q      Max
-3.1620  -0.3646   0.4394   0.5750   1.9511

Coefficients: (2 not defined because of singularities)
Estimate Std. Error z value Pr(>|z|)
(Intercept)     -8.59629    2.24708  -3.826  0.00013 ***
  age           -0.02051    0.02952  -0.695  0.48727
quiz            0.40771    0.14342   2.843  0.00447 **
  GenderBinary     0.20075    0.31010   0.647  0.51738
AfAm             0.12910    1.46380   0.088  0.92972
Asian            0.30368    1.38576   0.219  0.82654
Indian           0.53739    1.70989   0.314  0.75331
Latinx           1.31899    1.48271   0.890  0.37369
White            0.97888    1.31305   0.746  0.45597
MENA                NA         NA      NA       NA
veterans        -0.21321    1.03200  -0.207  0.83633
BE              -0.43124    0.38350  -1.124  0.26081
FE                  NA         NA      NA       NA
enrollments      1.32030    0.14001   9.430  < 2e-16 ***
  int_logic_score  0.16522    0.10418   1.586  0.11276
---
  Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 530.29  on 436  degrees of freedom
Residual deviance: 341.12  on 424  degrees of freedom
(157 observations deleted due to missingness)
AIC: 367.12

Number of Fisher Scoring iterations: 5
```

Then broke it down further into separate racial categories. This way I could compare specific groups as well as look at the larger group of People of Color vs. White students.

```
> log.reg.white <-
glm(graduated~age+quiz+GenderBinary+White+veterans+BE+int_logic_score+enrollm
ents, data=alldata_safe1, family="binomial")
> log.reg.white

Call:  glm(formula = graduated ~ age + quiz + GenderBinary + White +
           veterans + BE + int_logic_score + enrollments, family = "binomial",
         data = alldata_safe1)

Coefficients:
  (Intercept)           age          quiz     GenderBinary          White
veterans
BE  int_logic_score      enrollments
-7.9440         -0.0219          0.4028          0.2002          0.4713
-0.3806
-0.4447          0.1636          1.3098

Degrees of Freedom: 436 Total (i.e. Null);  428 Residual
(157 observations deleted due to missingness)
Null Deviance:              530.3
Residual Deviance: 343.2  AIC: 361.2
> summary(log.reg.white)

Call:
  glm(formula = graduated ~ age + quiz + GenderBinary + White +
        veterans + BE + int_logic_score + enrollments, family = "binomial",
      data = alldata_safe1)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-3.2680  -0.3566   0.4492   0.5757   1.8671

Coefficients:
  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -7.94399    1.83012  -4.341 1.42e-05 ***
  age           -0.02190    0.02931  -0.747  0.45509
quiz            0.40282    0.14299   2.817  0.00485 **
  GenderBinary     0.20024    0.30797   0.650  0.51558
White           0.47126    0.35764   1.318  0.18761
veterans        -0.38063    0.99086  -0.384  0.70088
BE             -0.44465    0.38311  -1.161  0.24578
```

int_logic_score 0.16355   0.10257   1.595 0.11080
*enrollments   1.30982   0.13857   9.453 < 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: 530.29 on 436 degrees of freedom
Residual deviance: 343.17 on 428 degrees of freedom
(157 observations deleted due to missingness)
AIC: 361.17

Number of Fisher Scoring iterations: 5

I created training and testing sets to test the viability of this data, with this result:
confusionMatrix(dataTest$graduated, predicted, threshold = 0.5)

	0	1
0	32	12
1	16	106

This particular logistic regression (test data) has a sensitivity rate of 75% and a specificity rate of 36%. Concordance is 84% and misclassification error rate is 12%. This is a good model based on those calculations.

Using this data, I calculated the following initial findings:

new.MoC.quiz <- data.frame(age=27, White=0, GenderBinary=1, quiz=c(4:8), veterans=0, BE=1, int_logic_score=13, enrollments=4)
> predict(log.reg.white, new.MoC.quiz, type = "response")
4 5 6 7 8
0.5491322 0.6456524 0.7316088 0.8030739 0.8591727

> new.WoC.quiz <- data.frame(age=27, White=0, GenderBinary=0, quiz=c(4:8), veterans=0, BE=1, int_logic_score=13, enrollments=4)
> predict(log.reg.white, new.WoC.quiz, type = "response")
4 5 6 7 8
0.4992324 0.5986270 0.6905229 0.7694810 0.8331614

> new.ytm.quiz <- data.frame(age=27, White=1, GenderBinary=1, quiz=c(4:8), veterans=0, BE=1, int_logic_score=13, enrollments=4)

```
> predict(log.reg.white, new.ytm.quiz, type = "response")
4         5         6         7         8
0.6611506 0.7448334 0.8136739 0.8672522 0.9071815

> new.ytw.quiz <- data.frame(age=27, White=1, GenderBinary=0, quiz=c(4:8),
veterans=0, BE=1, int_logic_score=13, enrollments=4)
> predict(log.reg.white, new.ytw.quiz, type = "response")
4         5         6         7         8
0.6149554 0.7049554 0.7813970 0.8424596 0.8888908
```

(edited column headings for clarity)

Results:
This shows some upsetting information, but information that can be valuable for us as an organization. White men and women have higher likelihoods of graduating across the board regarding initial quiz scores than men and women of color. Men of each subgroup have higher likelihoods of graduating than women in their group.

Additionally, every prediction model answers question 5 above with a resounding yes, higher logic scores correlate positively with a student's likelihood of graduating.

However, I wanted to look at it without the extraneous variables, so I removed age, enrollments, and int_logic_score in order to expand the field. This increased the AIC greatly, and gave me lower percentages, but similarly ranked predictions. This is an example where I could keep iterating and removing variables and find more and more information, so I decided to stop with this information and move on to the next question.

Recommendations for Turing

1. Continue using the logic test as an admissions tool. This is strongly correlated with student success and likelihood to graduate. Require all applicants to take the logic test, refining the Fast Track process to include the logic test.
2. Offer support immediately to students who score a 4 or 5 on their initial logic evaluation.
3. Collect anecdotal, qualitative data from students of color and female students who graduated and those who did not in order to deepen the analysis and determine why students from underrepresented groups are less likely to graduate.

Ideas for Further Research

Idea 1: For the second prediction model, I wanted to answer the following question: *Are there factors which make a student more likely to repeat?*

```
> logreg.repeats <- glm(formula = repeatbin ~ age + quiz + GenderBinary + White +
veterans + BE + int_logic_score, family = "binomial", data = alldata_safe1)
> logreg.repeats

Call:  glm(formula = repeatbin ~ age + quiz + GenderBinary + White +
    veterans + BE + int_logic_score, family = "binomial", data = alldata_safe1)

Coefficients:
    (Intercept)              age             quiz     GenderBinary            White
veterans           BE  int_logic_score
       -1.08981          0.05289         -0.27543          0.24711         -0.75984
-0.69285          1.01557         -0.04254

Degrees of Freedom: 436 Total (i.e. Null);  429 Residual
  (157 observations deleted due to missingness)
Null Deviance:              400.7
Residual Deviance: 379.9  AIC: 395.9
> summary(logreg.repeats)

Call:
glm(formula = repeatbin ~ age + quiz + GenderBinary + White +
    veterans + BE + int_logic_score, family = "binomial", data = alldata_safe1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.1489  -0.6219  -0.5347  -0.3830   2.3462

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.08981    1.54859  -0.704   0.4816
age              0.05289    0.02740   1.931   0.0535 .
quiz            -0.27543    0.13077  -2.106   0.0352 *
GenderBinary     0.24711    0.30073   0.822   0.4113
```

```
White           -0.75984    0.30071  -2.527   0.0115 *
veterans         -0.69285    1.13143  -0.612   0.5403
BE              1.01557    0.44205   2.297   0.0216 *
int_logic_score -0.04254    0.08563  -0.497   0.6193
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 400.69  on 436  degrees of freedom
Residual deviance: 379.94  on 429  degrees of freedom
  (157 observations deleted due to missingness)
AIC: 395.94

Number of Fisher Scoring iterations: 5
```

This is initially very interesting, showing a significant relationship between race, quiz score, program, and age with a person’s likelihood of repeating.

Idea 2: Further research that would be valuable for our admissions process would be to review the success rates of various interviewers (question 4 above). This would require some data wrangling to make the list of interviewers into a readable format for logistic or linear regression. I would probably make new binary columns or assign each interviewer a number then run a similar regression model to the one above.

Idea 3: Create a linear regression model by scaling the ‘behavior’ column from 1-5, then backtracking that to interview red flags (this would require significantly more data collection than is currently available).