PREDICTING GENDER IN INTRODUCTORY CS

PROJECT OVERVIEW

As I have often said, one of the biggest challenges of this century is successfully getting people into the technical workforce. We are now in the age of automation with autonomous cars driving down our streets and Alexa and Siri becoming an extension of our homes. With the increase of these systems, it is likely that we are also going to see an increase in technical jobs.

This shift in the workforce towards highly skilled, technical knowledge workers poses a challenge on the supply side; mostly because of a lack of presence of computer science in K-12 education; the underproduction of post-secondary degrees in computer science; the underrepresentation of women and the underrepresentation of ethnic minorities.

One of the solutions that have been proffered for this problems is redesigning introductory computer science to broadening participation.

As part of my doctoral study, I decided to study the socio-curricular factors that affect the decision to participate in introductory computer science through a data-driven lens. To do this, I designed a research study investigating the role of computer science self-identity centered around the experiences of undergraduates in two introductory computer science classes at UC Berkeley.

PROBLEM STATEMENT

With this project, the problem I am interested in investigating is the gendered experience of these the two CS classes in my study. Using machine learning algorithms, I want to predict and identify the most salient variables that govern the experience of men versus women in introductory CS at an elite research university like Berkeley.

To predict gender in intro CS at Berkeley we suggest the following strategy:

(a) Explore the dataset to ensure its integrity and understand the context.

(b) Identify features that may be used. If possible, engineer features that might provide greater discrimination.

(c) With the understanding that this a "classification" task, explore a couple of classifiers that might be well suited for the problem at hand.

(d) Once a classifier has been selected, tune it for optimality.

METRICS

Predicting gender in intro CS is a supervised learning problem. To determine the performance of the model, I have selected “accuracy" as the metric of most interest. In addition to that, we will take a look at the confusion matrix for the output of each model to give us more insight into how good our classifiers are at discriminating the data based on gender.

ANALYSIS

DATASET

This project uses a dataset which I created as part of my doctoral research. The dataset consists of survey responses. A copy of the survey instrument can be found in the appendix of this report . The survey instruments were developed to measure participants' self-reported efficacy along several dimensions.

(a) Self-reported attitudes about CS

(b) Gendered belief about CS ability

(c) Career driven beliefs about CS

(d) Self-reported attitudes about computational thinking

(e) Self-reported attitudes about CS class belonging

(f) Self-reported beliefs about collegiality

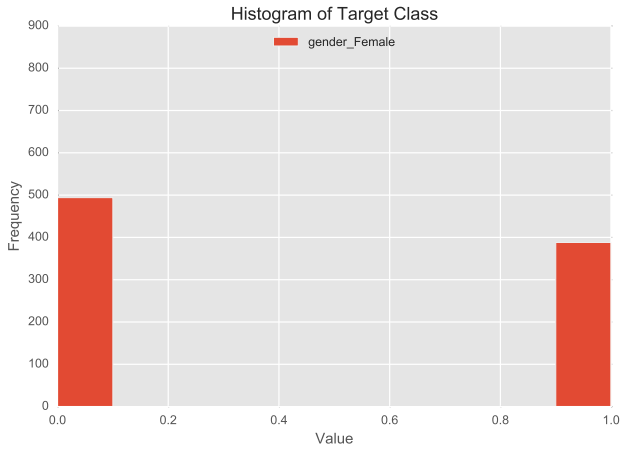
(g) Prior collegiate CS exposure

(h) CS mentors and role models

Majority of the questionnaire uses a 5-point Likert scale (where 1 = Disagree, 3 = Neutral and 5 = Agree). A code book was created to facilitate ease of analysis and interpretability of results. Using the function `dataDescr()` gives an introduction to the dataset along with codes and the questions which those code represent. For individual look up of codes, the function `dataLookUp('atct_8')` can be used.

The dataset consists of 882 instances with no missing data. Further, there are 494 males and 388 female samples in the dataset.

An interesting aspect of this dataset is that the class labels for our classification are slightly unbalanced at a ratio of around 1:1.2 for male students as can be seen in figure 0.1a.



(a)

Figure 0.1: **Gender plot.** *The histogram shows a slightly unbalanced target dataset with 494 values of {0: male} and 388 values of {1: female}.*

ALGORITHMS AND TECHNIQUES

For the problem of predicting gender in intro CS I experimented with four different classifiers, a decision tree classifier, two ensemble methods and a support vector machine:

(a) A RandomForestClassifier
I selected this learner because it is considered one of the best off-the-shelf learning algorithm, and requires almost no tuning.

(b) An eXtreme Gradient Boosted (XGBoost) Trees Classifier
XGBoost is an advanced implementation of gradient boosting algorithm. From reading literature on machine learning in practice, the XGBoost classifier has differentiated itself as a classifier that has successfully demonstrated its performance in a wide range of problems from particle physics, to ad click-through rate prediction and so on. For example, "among the 29 challenge winning solutions published at Kaggle's blog during 2015, 17 solutions used XGBoost."

(c) Support Vector Machine (SVMs)
I selected the SVMs because they are very robust classifiers and *more importantly*, they have a method of *biasing* the soft-margin constant, C, to correct for class imbalances.

(d) Decision Tree Classifier
The *major* reason why the decision tree classifier was selected was its interpretability. For this problem domain, it is not just satisfactory to discriminate between male and female students, what learning researchers ultimately want is to gain *insights* into

what the salient factors around the experience of intro CS are so they can correct for negative outcomes.

BENCHMARK

This is novel research, as a result, there are no benchmarks we can compare the performance of our classifiers with.

METHODOLOGY

DATA PREPROCESSING

To prepare our data for classification, we need to devise a scheme to transform all features into numeric data. This dataset as several non-numeric columns that need converting. Many of them are simply `yes` and `no`, e.g. `prcs_2`. We can reasonably convert these into '1'/'0' (binary) values. For the columns whose values are 'Nan', we will convert these to '0'. We removed the spaces from some column names with the understanding that the tree plotting algorithm for Xgboost will fail if column names have spaces.

We scaled the features using a minimax scaler to get better output for our SVM. This yielded the follow:

- Disagree = 0.0
- Somewhat disagree = 0.2
- Neutral = 0.6
- Somewhat agree = 0.8
- Agree = 1.0

FREQUENCY DISTRIBUTION

We created a frequency distribution for some dimensions in our data to see if there are features that have extremely low variability in their distribution. From figures 0.3, 0.2, and 0.4, we know that these variables are broadly distributed.

From 0.5, we can see that the distribution for the dimension `atcsgender` is extremely skewed to the right, further we notice that `atcsgender_2` is bimodal.

In doing these frequency distributions we are trying to gain an understanding of the variables and determine if we need to reject some of them, or collapse other.

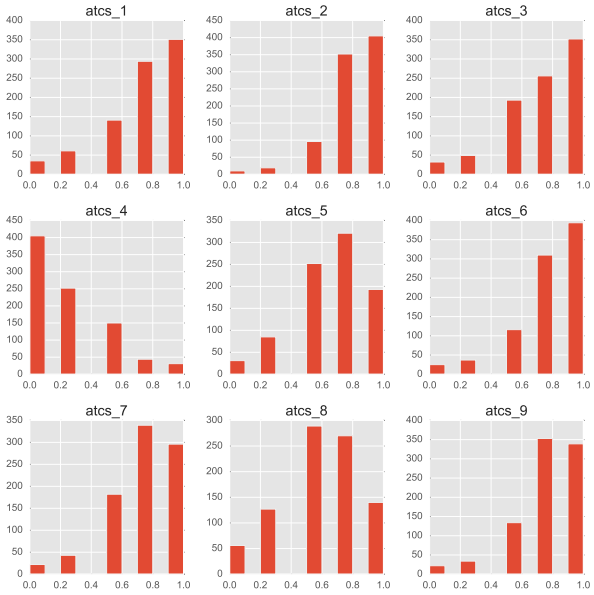


Figure 0.2: **Frequency distribution for dimension atcs.** *Self-reported attitudes about CS.*

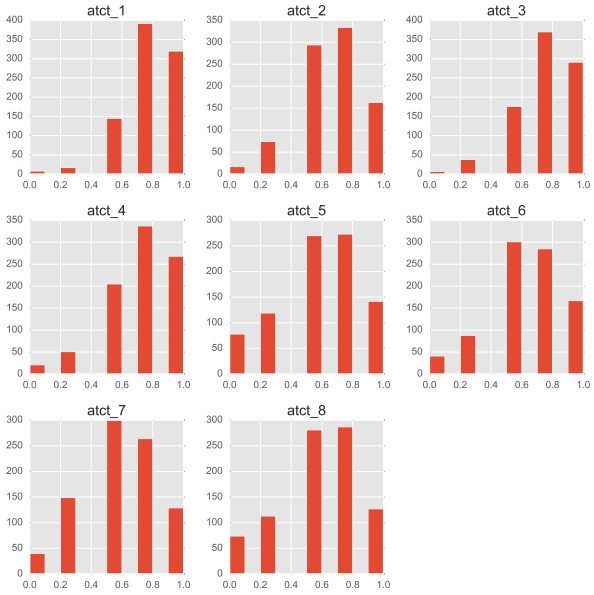


Figure 0.3: **Frequency distribution for dimension atct.** *Self-reported attitudes about computational thinking.*

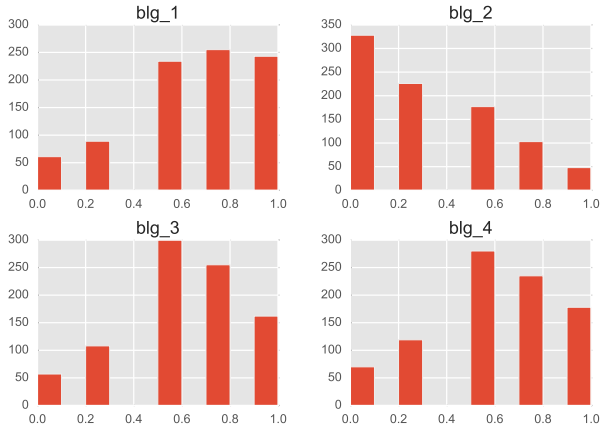


Figure 0.4: **Frequency distribution for dimension blg.** *Self-reported attitudes about CS class belonging.*

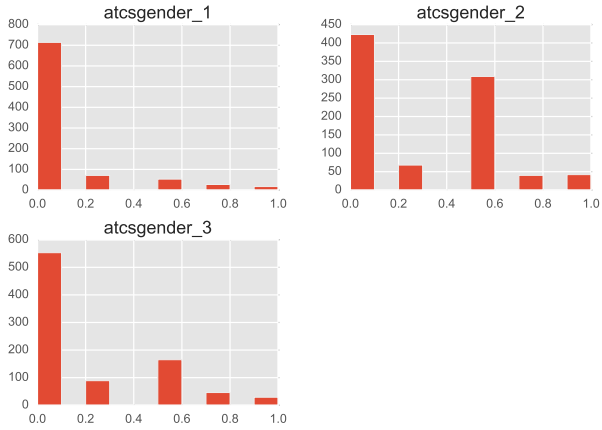


Figure 0.5: **Frequency distribution for dimension atcsgender.** *Gendered belief about CS ability.*

IMPLEMENTATION

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

- Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?
- Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?
- Was there any part of the coding process (e.g., writing complicated functions) that should be documented?

REFINEMENT

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

- Has an initial solution been found and clearly reported?
- Is the process of improvement clearly documented, such as what techniques were used?
- Are intermediate and final solutions clearly reported as the process is improved?

RESULTS

(approximately 2 - 3 pages)

MODEL EVALUATION AND VALIDATION

When I did feature selection using SelectPercentile on top 10% of the features, the model over-fitted, giving a 100% score on all the three classifiers used.

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model,Äôs solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

- Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?
- Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?
- Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?
- Can results found from the model be trusted?

JUSTIFICATION

In this section, your model,Äôs final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

- Are the final results found stronger than the benchmark result reported earlier?
- Have you thoroughly analyzed and discussed the final solution?
- Is the final solution significant enough to have solved the problem?

CONCLUSION

(approximately 1 - 2 pages)

FREE-FORM VISUALIZATION

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- Have you visualized a relevant or important quality about the problem, dataset, input data, or results?
- Is the visualization thoroughly analyzed and discussed?
- If a plot is provided, are the axes, title, and datum clearly defined?

REFLECTION

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- Have you thoroughly summarized the entire process you used for this project?
- Were there any interesting aspects of the project?
- Were there any difficult aspects of the project?
- Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?

IMPROVEMENT

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to

make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current sLike Sita's family, mine where non-pleased with my choice. olution. Questions to ask yourself when writing this section:

- Are there further improvements that could be made on the algorithms or techniques you used in this project?
- Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?
- If you used your final solution as the new benchmark, do you think an even better solution exists?

SURVEY INSTRUMENTS

DEMOGRAPHICS

- Year in university [Freshman, Sophomore, Junior, Senior]
- Gender [Male, Female, Other]
- What is your reason for taking this class [interested, other]
- What is your major?

ATTITUDES TOWARDS COMPUTER SCIENCE

- I like to use Computer Science to solve problems.
- Knowledge of computing will allow me to secure a good job.
- I can learn to understand computing concepts.
- My career goals do not require that I learn computing skills.
- I can achieve good grades (C or better) in computing courses.
- I do not like using computer science to solve problems.
- I am confident that I can solve problems by using computer applications.
- The challenge of solving problems using computer science appeals to me.
- I am comfortable with learning computing concepts.
- I would take additional Computer Science courses if I were given the opportunity.
- I am confident about my abilities with regards to computer science.
- I do think I can learn to understand computing concepts.

ATTITUDES ABOUT COMPUTATIONAL THINKING

- I am good at solving a problem by thinking about similar problems I've solved before.
- I have good research skills.
- I am good at using online search tools.
- I am persistent at solving puzzles or logic problems.

- I know how to write computer programs.
- I am good at building things.
- I'm good at ignoring irrelevant details to solve a problem.
- I know how to write a computer program to solve a problem.
- I work well in teams.
- I think about the ethical, legal, and social implications of computing.

COMPUTER SCIENCE MENTORS AND ROLE MODELS

- Before I came to UC Berkeley, I knew people who have careers in Computer Science.
- There are people with careers in Computer Science who look like me.
- I have role models within the Computer Science field that look like me.

IDENTITY AND SELF EFFICACY

- In this class, I feel I belong.
- In this class, I feel awkward and out of place
- In this class, I feel like my ideas count
- In this class, I feel like I matter.
- I am comfortable interacting with peers from different backgrounds than my own (based on race, sexuality, etc.)
- I have good cultural competence, or the ability to interact effectively with people from diverse backgrounds.
- Our class materials (e.g., case studies and projects) were relevant and practical

GENDERED BELIEF ABOUT COMPUTER SCIENCE ABILITY

- Women are less capable of success in CS than men
- Men have better math and science abilities than women.
- Women are smarter than men.

- Did you take a CS course in High School?
- Did you have exposure to Computer Science before UC Berkeley?
- Did a family member introduce you to Computer Science?
- Did you have a close family member who is a Computer Scientist or is affiliated with computing industry?
- Did your school offer AP CS?
- How prepared did you feel about this class before it started?
- Will you be taking any more CS classes (if so which ones?)
- (For 61A only) Have you taken CS10, The Beauty and Joy of Computing?