PREDICTING GENDER IN INTRODUCTORY COMPUTER SCIENCE (CS)

PROJECT OVERVIEW

One of the biggest challenges of this century is successfully getting people into the technical workforce. We are now in a new technological era with autonomous cars driving down our streets and bots like Alexa and Siri becoming an extension of our lives. As automation continues to gain ground, so too are the new industries it helps to create. This new era is creating a new kind of worker, the highly-skilled knowledge worker, in particular, the highly-skilled *technology* knowledge worker.

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 solution that has been proffered for this problem 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.

The major motivation for doing this project is to gain some insight into what factors most govern the experiences of historically underrepresented students as they enter the CS pipeline at the University level. For this particular study, the URMs are the 388 female students who are participating in these classes, for many of them, this represents their first experience with CS.

If we are able to truly understand what factors specifically *increase* a **sense of belonging in CS** for URMs, then we can create environments where these factors abound. Once we have such an environment; we can determine the effect on both male and female students; if we find out that these kinds of environment positively support CS belonging in both groups, we can then make the recommendation that we design our CS learning environments along those lines. The first step in such a scheme would be to predict the salient factors around the gendered experience of CS, which is what this reason why this project exists.

PROBLEM STATEMENT

With this project, the problem I am interested in investigating is the gendered experience of the two CS classes in the 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 I will be using the following problem-solving strategy that is common to most machine learning challenges:

(a) Explore the dataset. Usually, I would explore the dataset to ensure its integrity and understand the context. But in this case, I will skip this step since I designed the study and collected the data, as such, I am well versed of the context. Further, I have done previous work on this dataset, so I know its boundaries.

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

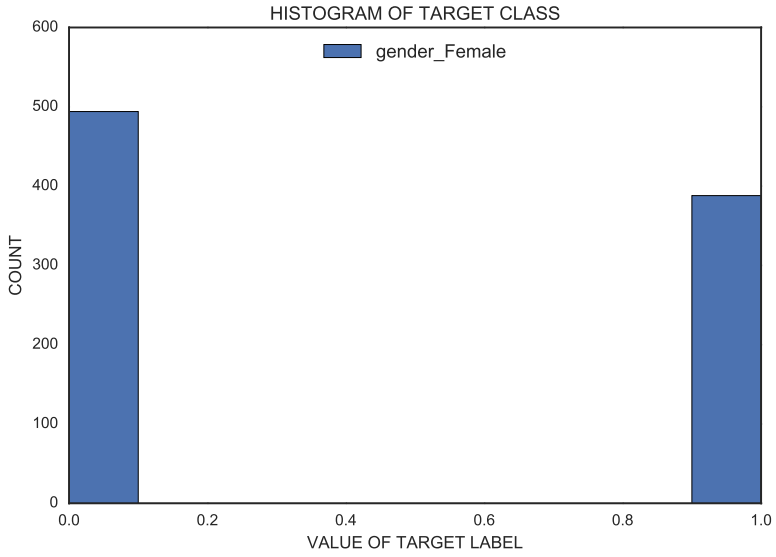
(c) With the understanding that this a "classification" task, explore a couple of classifiers that might be well suited for the problem at hand.

(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 will be using the F_1 score, i.e., the weighted average of precision and recall as our metric of choice. I am choosing to use the F_1 score as my metric over the accuracy score because of the label imbalance in the dataset. Figure 0.1 shows a histogram of the class label. This dataset has a gender variable which we have expanded into two dummy variables: 'gender_Female' and 'gender_Male.' From this histogram, one can see there is a slight imbalance in the target labels in favor of males, with a ratio of 1.2.

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



ANALYSIS

DATASET

The dataset used in this project 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 attitudes along several dimensions:

(a) CS beliefs

(b) Gendered belief about CS ability

(c) Career driven beliefs about CS

(d) Computational thinking beliefs

(e) CS belonging

(f) Collegiality

In addition, the survey also collected data around student background:

(a) Prior collegiate CS exposure

(b) CS mentors and role models

(c) University demographics

Majority of the questionnaire uses a 5-point Likert scale (where 1 = Strongly Disagree, 3 = Neutral and 5 = Strongly Agree). A code book was created to facilitate ease of analysis and interpretability of results. The dataset consists of 882 instances with no missing data. Further, there are 494 males and 388 female samples in the dataset.

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) I selected a Random Forest classifier because it is considered one of the best off-the-shelf learning algorithm, and requires almost no tuning.

(b) Another selection was the eXtreme Gradient Boosted (XGBoost) trees classifier; which is an advanced implementation of the 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. For example, "among the 29 challenge winning solutions published at Kaggle's blog during 2015, 17 solutions used XGBoost."

(c) The Support Vector Machine (SVMs) was selected because they are very robust classifiers and *more importantly*, they have a method to correct for class imbalances.

(d) Finally a Decision Tree classifier was also selected. 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 I ultimately want is to gain *insights* into what the salient factors around the experience of intro CS are, based on gender.

BENCHMARK

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

METHODOLOGY

DATA PREPROCESSING

To prepare the data for classification, all features need to be transformed into numeric data. This dataset has several non-numeric columns that need converting. Many of them take on `yes` and `no` values, e.g. `prcs_2`. These can be reasonably convert these into '1'/'0' (binary) values. For the columns whose values are 'Nan', these will be converted to '0'. Further, spaces will be removed from column names with the understanding that the tree plotting algorithm for Xgboost will fail if column names have spaces.

The feature were scaled using a minimax scaler to get better output for our SVM. This yielded the following values:

- Strongly Disagree = 0.0
- Disagree = 0.2
- Neutral = 0.6
- Agree = 0.8
- Strongly 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 spread in their distribution. From figures 0.4, 0.3, and 0.5, we know that these variables are broadly distributed.

From 0.2, 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 atcsgender.** *Gendered belief about CS ability.*

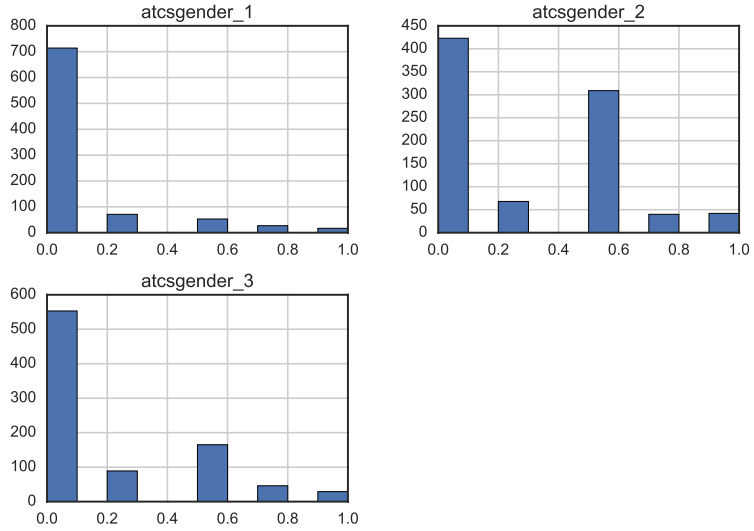


Figure 0.3: **Frequency distribution for dimension atcs.** *Self-reported attitudes about CS.*

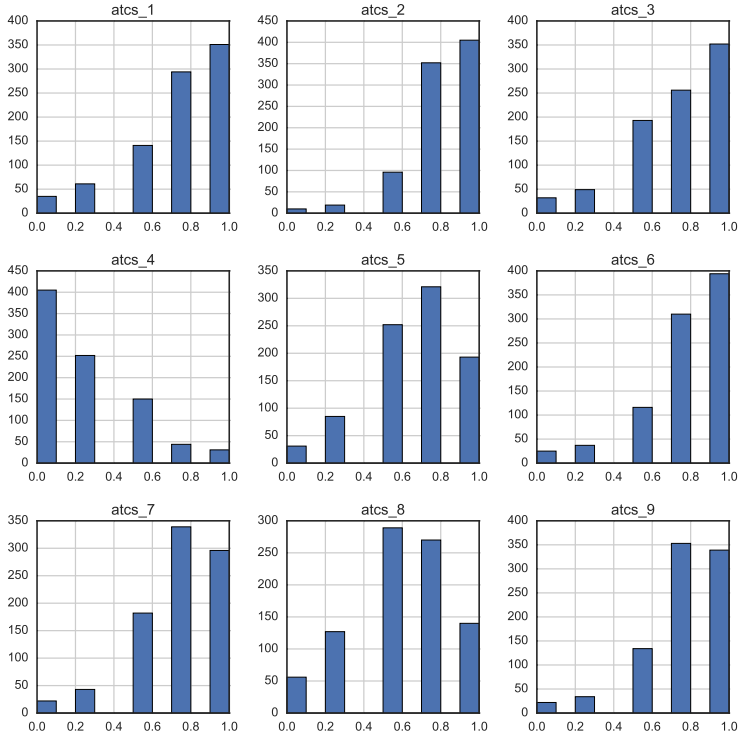


Figure 0.4: **Frequency distribution for dimension atct.** *Self-reported attitudes about computational thinking.*

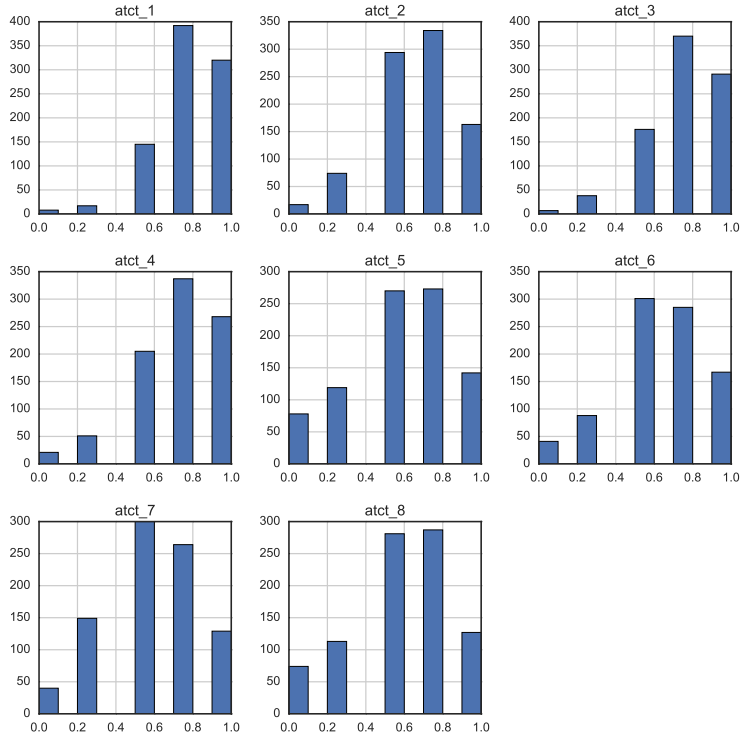
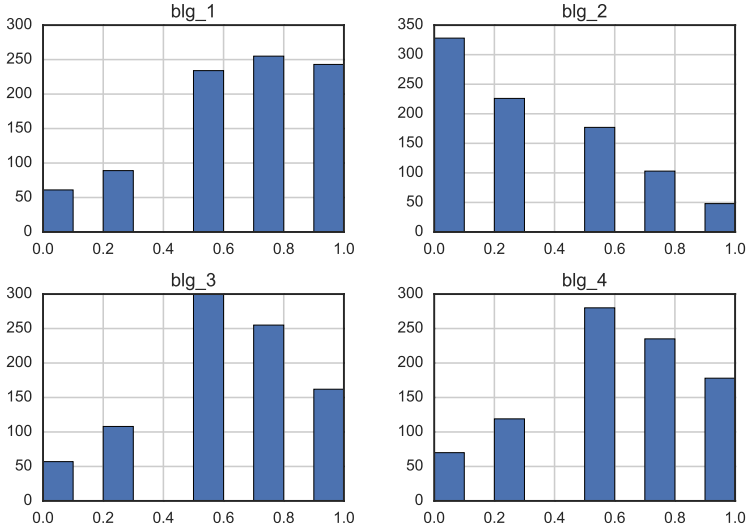


Figure 0.5: **Frequency distribution for dimension blg.** *Self-reported attitudes about CS class belonging.*



We implemented the four learning algorithms. For each of the learners we implemented the baseline algorithm using a stratified shuffle split cross validation with 50 folds and calculated the F_1 scores and looked at the confusion matrices respectively.

Table 0.1: Scores

Result of training the baseline classifiers	
Classifier	F1 Score
SVC	0.541%
DecisionTree	0.579%
RandomForestClassifier	0.608%
XGBClassifier	0.693%

In figure 0.6 we see the decision tree for the first two xgboost trees. This figure gives us insight into which features were doing the most work of splitting the data, and consequently may have the largest impact on predicting gender in introductory CS experience.

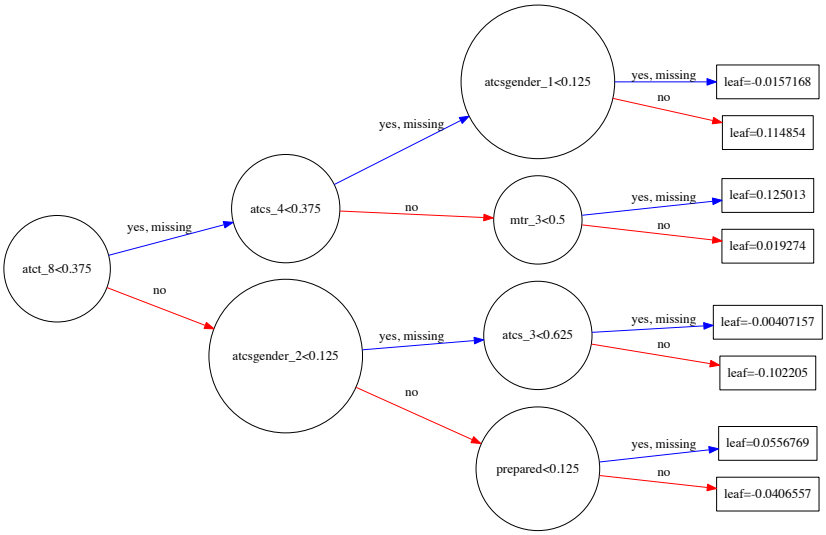
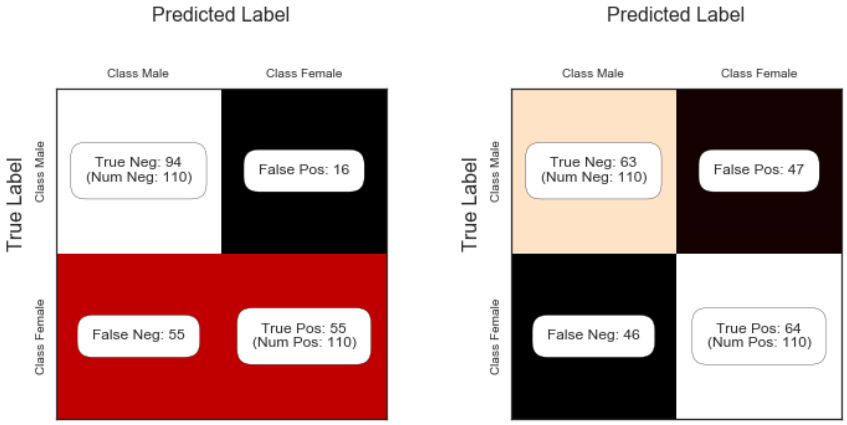
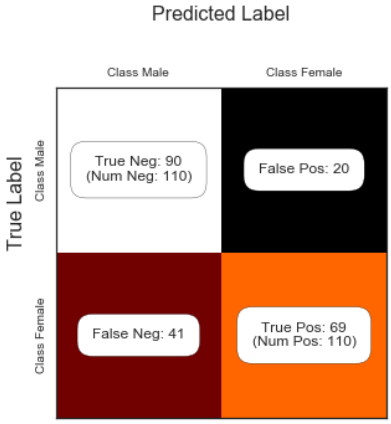


Figure 0.6: **XgBoost estimator decision tree.**

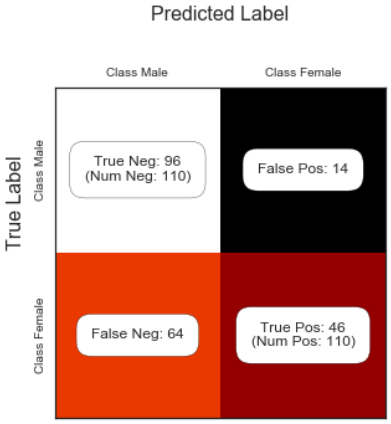


(a) Random Forest

(b) Decision Tree



(c) XgBoost



(d) SVC

Figure 0.7: **Confusion Matrices of Baseline Classifiers**

RESULTS

MODEL EVALUATION AND VALIDATION

We tuned our model using sklearn's `GridSearch` in conjunction with a {k=50 fold} `StratifiedShuffleSplit` function, we tuned both the tree parameters as well as the task parameters of our XGBoost classifier to control for over-fitting and improve over all performance. The parameters we tuned are as follows:

- Parameters for Tree Booster
  - `max_depth`
    - Maximum depth of tree
    - Range $[1, \infty]$, default 6, tuned on $[4, 6, 8, 10]$
  - `n_estimators`
    - Minimum number of trees
    - Range $[2, \infty]$ default 2, tuned on $range(100, 1100, 100)$
- Task Parameter
  - `learning_rate`
    - Scale the contribution of each tree by learning rate
    - Range $[0, 1]$, tuned on $[0.2222, 0.4444, 0.6666, 0.8888]$

Once we performed our search through the hyper-parameter space to find the combination of hyper-parameters that maximized the performance of our classifier, we were able to improve the previous F_1 score by 0.027%. Here is the final model for classifying gender in introductory CS.

```
XGBClassifier(base_score=0.5, colsample_bylevel=1, colsample_bytree=0.6,
       gamma=0, learning_rate=0.2222, max_delta_step=0, max_depth=6,
       min_child_weight=1, missing=nan, n_estimators=600, nthread=-1,
       objective='binary:logistic', reg_alpha=0, reg_lambda=1,
       scale_pos_weight=1, seed=0, silent=1, subsample=0.7)
```

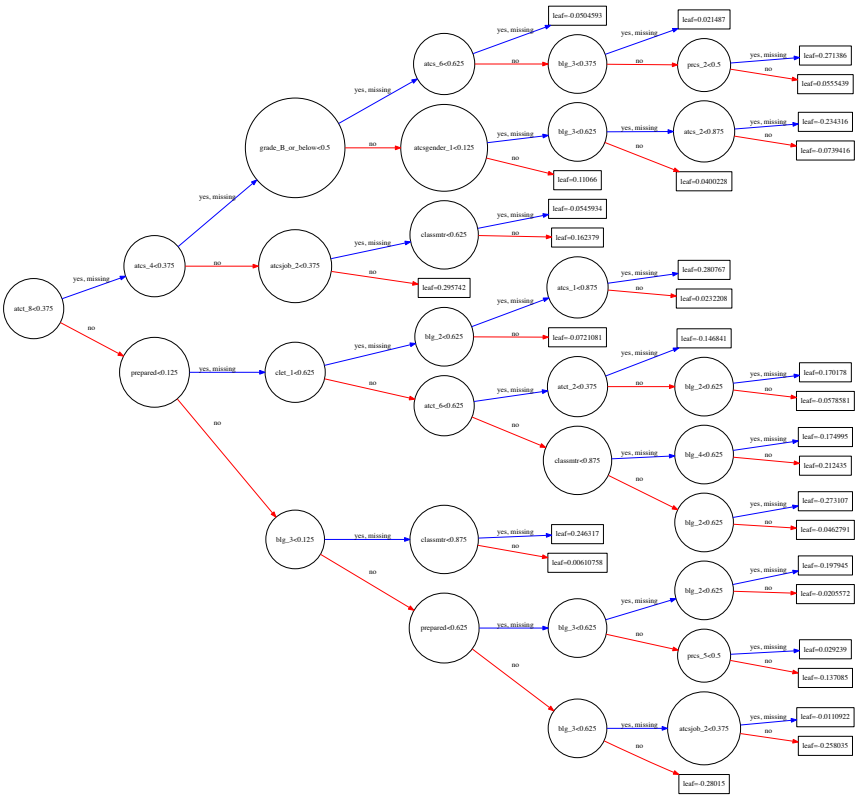


Figure 0.8: **Tuned XgBoost estimator decision tree.**

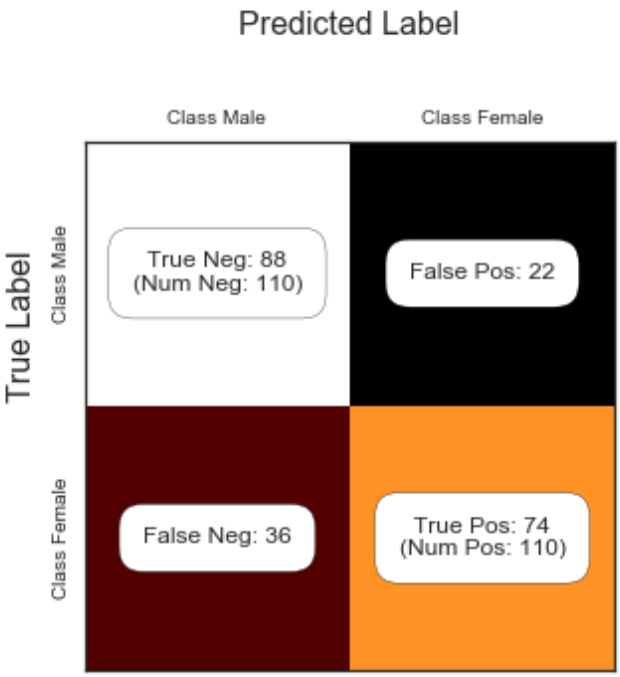


Figure 0.9: **Tuned XgBoost model confusion matrix.**

CONCLUSION

REFLECTION

SURVEY INSTRUMENTS

DEMOGRAPHICS

- 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?