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, we will be using the F_1 score, i.e., the weighted average of precision and recall as our metric of choice. We are making this choice over accuracy because of the label imbalance in our data. 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

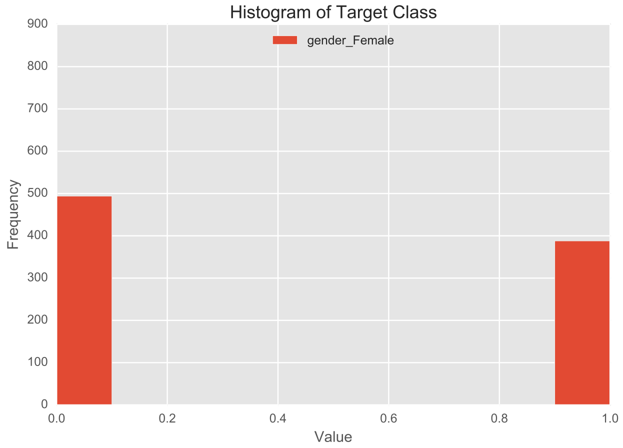
(g) Prior collegiate CS exposure

(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 following values:

- 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.

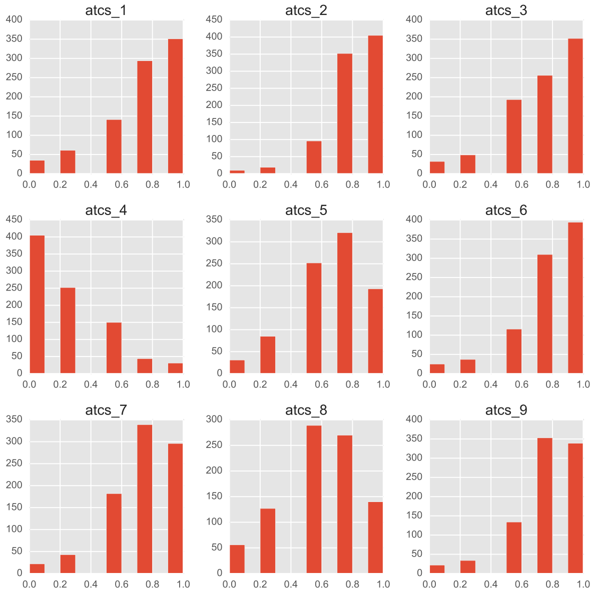


Figure 0.2: **Frequency distribution for dimension atcs.** *Self-reported attitudes about CS.*

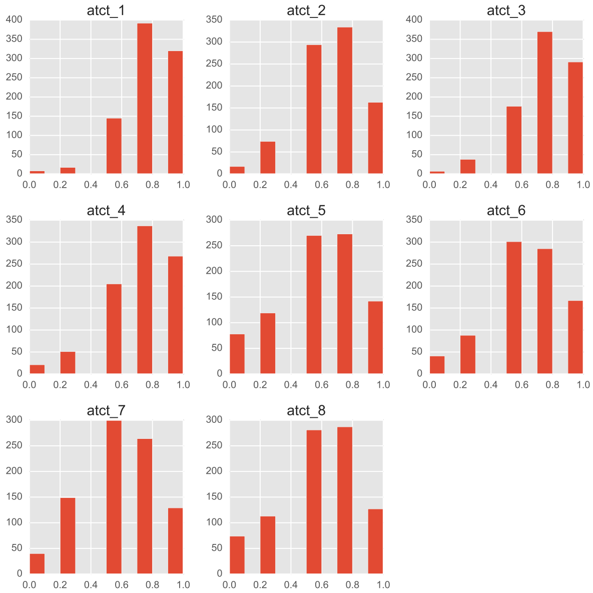


Figure 0.3: **Frequency distribution for dimension atct.** *Self-reported attitudes about computational thinking.*

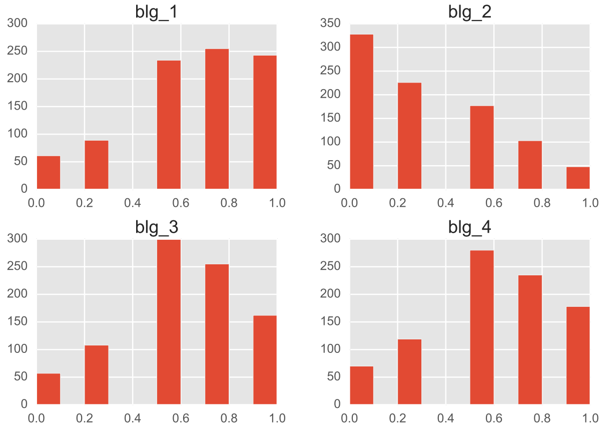


Figure 0.4: **Frequency distribution for dimension blg.** *Self-reported attitudes about CS class belonging.*

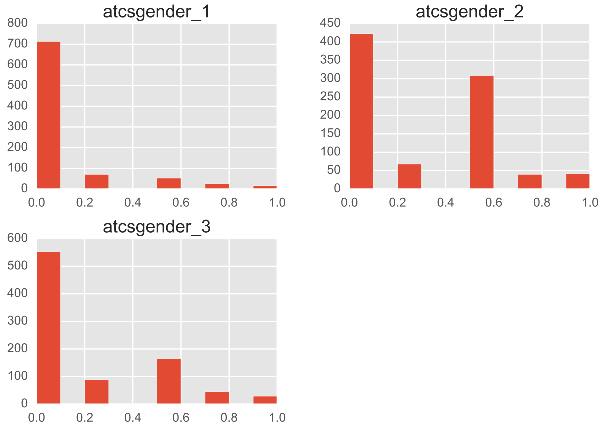


Figure 0.5: **Frequency distribution for dimension atcsgender.** *Gendered belief about CS ability.*

We implemented the four learning algorithms. For each of the learners we implemented the baseline algorithm using a stratified shuffle split cross validation with a thousand 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.

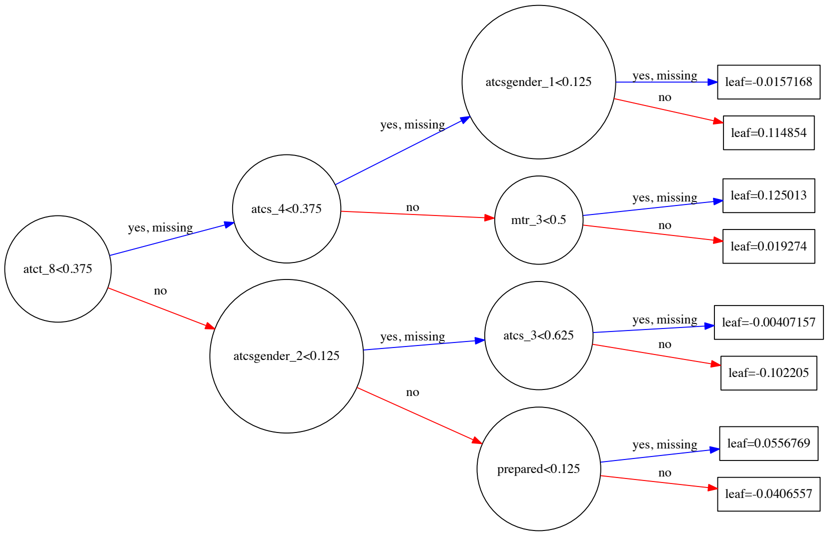
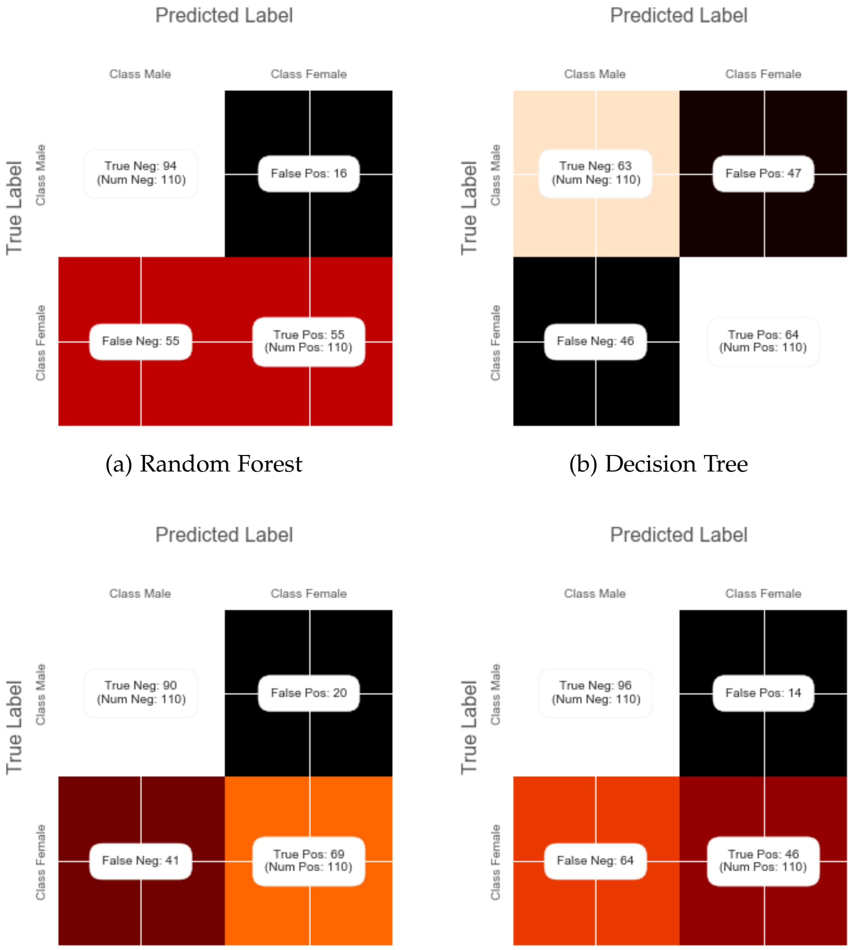


Figure 0.6: **XgBoost estimator decision tree.**



(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, ∞], 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)
```

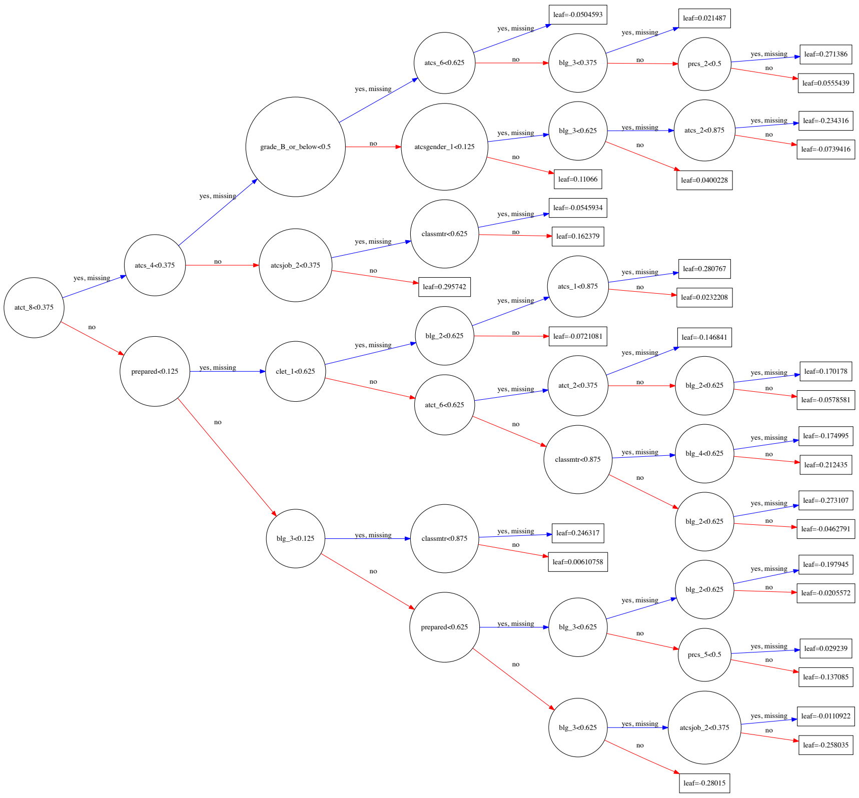


Figure 0.8: **Tuned XgBoost estimator decision tree.**

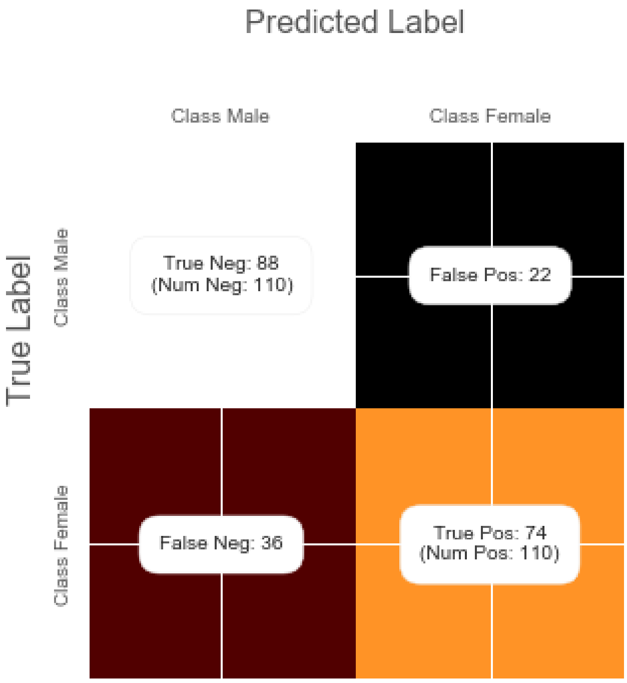


Figure 0.9: **Tuned XgBoost model confusion matrix.**

CONCLUSION

REFLECTION

SURVEY INSTRUMENTS

DEMOGRAPHICS

- 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.

Pre-collegiate CS preparation

- 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?