IDENTIFY FACTORS THAT PREDICT INTRO CS EXPERIENCE BASED ON GENDER

PROJECT OVERVIEW

I once listened to an episode of Star Talk Radio with Neil deGrasse Tyson titled "The Future of Humanity with Elon Musk." Ten minutes into the interview, Musk talks about having sophomoric philosophical wanderings as a student in college. He spent his time musing about the five things that would MOST affect the future of humanity. He thought they were the Internet, sustainable energy, artificial intelligence, rewriting human genetics, and space exploration. Bill Nye, who was also on the show suggested that he would add one more, "the education of women and girls."

Before I delve into the issue of educating women and girls, I will make a quick detour into the importance of retraining. What are my talking about? Am talking about **self-driving trucks**!!! Recently, Uber debuted its truck which autonomously drove across the country on a beer run. Like all the other techies out there, I was thrilled!!!!! But then the reality set in. I thought we would have a generation before this piece of AI would go into production. I thought there would be enough time for us to figure out how to retrain adults en masse for the new economy jobs. Turns out that wasn't true. The future is here.

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/or the underrepresentation of ethnic minorities. Which brings me back to Bill Nye the science guy.

I think of this problem as a big-data opportunity where we can kill two birds with one stone. We can invent adult education for workforce readiness en masse while leveraging that opportunity to equalize participation.

As Internet adoption increases, so too will be the opportunity to leverage online education to close the gap between the genders, particularly in emerging countries. A solid understanding of the factors that determine women's participation in computer science can help guide how we design these future learning environments. This project is the start of my journey into understanding those factors.

As part of my doctoral study, I decided to investigate 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 examining the role of computer science self-identity centered around the experiences of undergraduates in two introductory computer science classes at UC Berkeley. Once that study was completed, I didn't stop; I decided to ask new questions of the data. Specifically *what were the leading factors that made female students choose intro CS*?

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 identify the leading indicators of the experience of belonging broken down by gender in introductory CS at an elite research university like Berkeley.

To solve this problem, I will undertake the following course of action:

(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) Select an appropriate classifier based on the evaluation metric and tune it for optimality.

(e) Extract the top features responsible for discriminating the data.

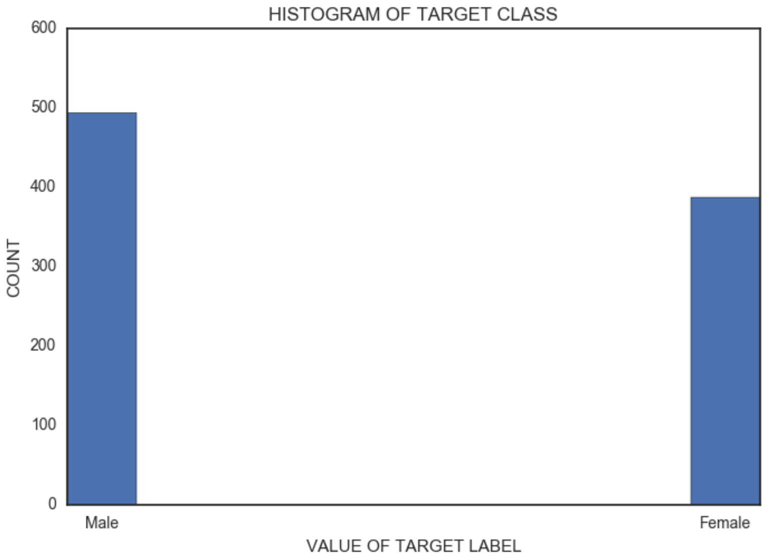
METRICS

The student dataset that I will be using for this project has unbalanced classes as can be seen in figure 0.1. It is important to pay attention to this class unbalance as it can cause the accuracy metric that is routinely used to judge classifiers to breakdown as the classes become more skewed.

Taking into account the unbalanced dataset, and the nature of the problem itself, instead of using accuracy as my evaluation criteria, I will use the **expected value** framework, 0.2. This framework allows me to evaluate each classifier with respect to the "potential" business value of the decisions it makes on the data. In real life scenarios, there is often a cost associated with misclassifying data. To really drive this idea home, permit me to take a detour into the world of credit assessment.

Machine learning applications are often used to determine whether a candidate should be given a loan or not. In this scenario, there is a real cost to a business if a good candidate is misclassified to be a risky candidate. When that happens, the business loses money it otherwise would have made from issuing the loan. So, in evaluating the

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

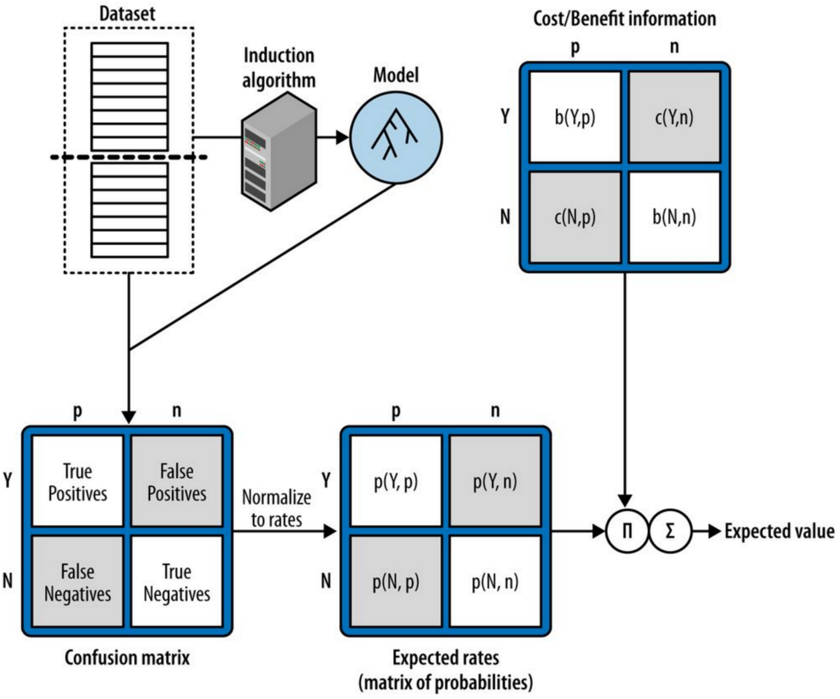


effectiveness of a classifier in separating such a dataset, we want to have a cost associated with a misclassification and a reward/benefit associated with a correct classification. A way to do that is to use the expected value framework whose equation is shown below:

$$\begin{aligned}\text{Expected Value} = {} & p(\mathbf{p}).[p(\mathbf{Y}|\mathbf{p}).b(\mathbf{Y},\mathbf{p}) + p(\mathbf{N}|\mathbf{p}).c(\mathbf{N},\mathbf{p})] \\ & + p(\mathbf{n}).[p(\mathbf{N}|\mathbf{n}).b(\mathbf{N},\mathbf{n}) + p(\mathbf{Y}|\mathbf{n}).c(\mathbf{Y},\mathbf{n})]\end{aligned} \tag{1}$$

This equation states that the expected value of a classifier is the expected rates multiplied by the cost/benefit value of each entry in the confusion matrix, weighted by the class priors. For this project, I have assigned a penalty of $c(\mathbf{N},\mathbf{p}) = -2$ for each false classification of the female class and a reward of $b(\mathbf{N},\mathbf{n}) = 5$ for each correct assignment. I came up with this numbers arbitrarily based on my desire to pick the classifier that best separated out the female class.

Figure 0.2: **Expected Value Calculation.** *Expected rates multiplied by the cost-benefit weighted by the class priors. Image from "Provost and Fawcett. Data Science for Business: What You Need to Know About Data Mining and Data-analytic Thinking, 2013."*



ANALYSIS

DATASET

The dataset I used in this project consists of survey responses. I developed the survey instruments to measure undergraduate students' self-reported attitudes along the following dimensions:

(a) `atcs`: CS beliefs

(b) `atcsgender`: Gendered belief about CS ability

(c) `atcsjob`: Career driven beliefs about CS

(d) `atct`: Computational thinking beliefs

(e) `blg`: CS belonging

(f) `cltrcmp`: Collegiality

In addition, I also collected data around students' background using the following dimensions.

(a) `prcs`: Prior collegiate CS exposure

(b) `mtr`: CS mentors and role models

(c) University demographics

Majority of the questionnaire I developed uses a 5-point Likert scale (where 1 = Strongly Disagree, 3 = Neutral and, 5 = Strongly Agree). I created a code book to facilitate my ease of analysis and the interpretability of results. The dataset consists of 45 features with 882 instances. Further, the dataset breakdowns into 494 male instances versus 388 female instances.

MISSING VALUES

The dataset with 45 features had two features with missing data.

- priorcs10: 43.88% missing
- reason_class: 0.68% missing

DATA PREPROCESSING

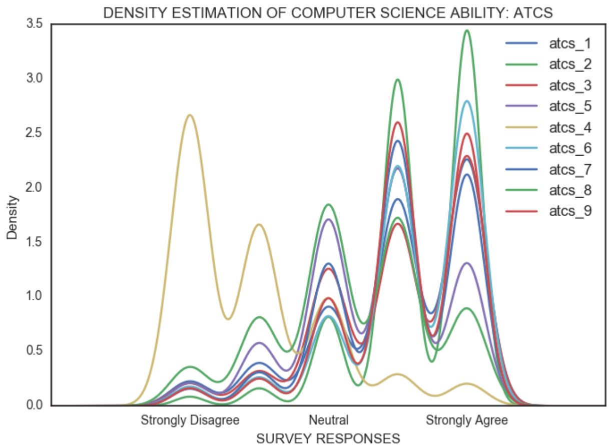
To prepare the data for classification, all features need to be transformed into numeric data. This dataset has several non-numeric columns that need converting. Many of them take on `yes` and `no` values, e.g. `prcs_2`. These can be reasonably convert these into '1'/'0' (binary) values. For the columns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

The features 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

summarizing the data

Figure 0.3: **Density estimation for dimension atcs.** *Self-reported attitudes about CS.*



I created a density estimation for some dimensions in the data to gain an rough understanding of student experiences. The distributions of most of the dimensions looked very similarly to that of 0.3. Most of the data is either skewed to the left or skewed to the right. As a result, I rejected using descriptive statistics to summarize the data in favor quantiles represented by box plots as can be seen in figure 0.4.

So what does figure 0.4 tell us about the data? From that figure, we can see that the median of this dimension is approximately at the 75 percentile, which based on our Likert scale dataset means most students generally agree with the mostly positive attitudinal questions asked about their CS beliefs. For computational thinking, from figure 0.5 we see that most of the data in this dimension follow a similar distribution.

From 0.6a, I can see that the distribution for the dimension `atcsgender` is really skewed to the right, i.e., most students *strongly disagree* with the statements. That does not come as a surprise, what I found fascinating is that the median for `atcsgender_2` is at the 25 percentile, which corresponds to "neutral." You can see this in the boxplot in figure 0.6b. While students do not agree that women are smarter than men, half of them are undecided about this statement!

Figure 0.4: **Quantiles for dimension** `atcs`. *Self-reported attitudes about CS.*

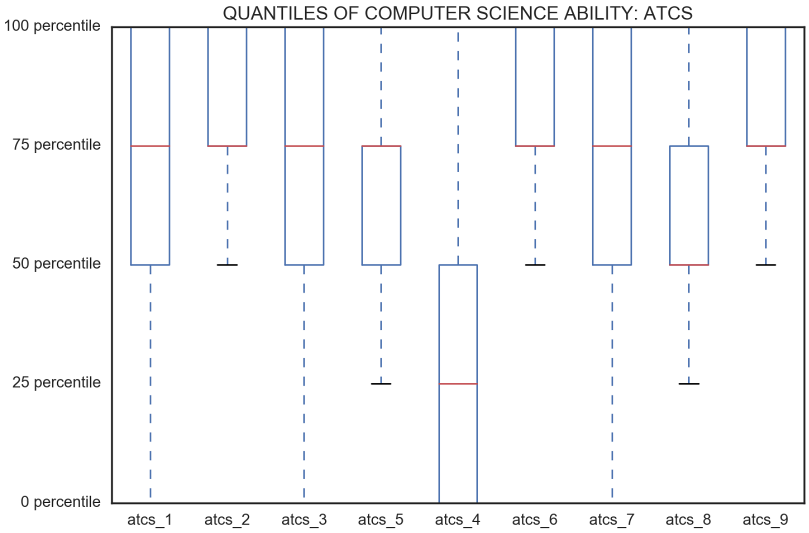
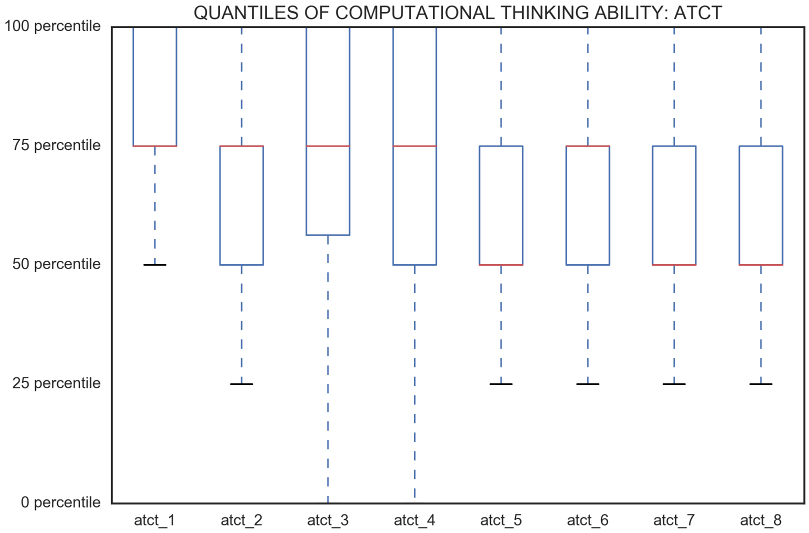


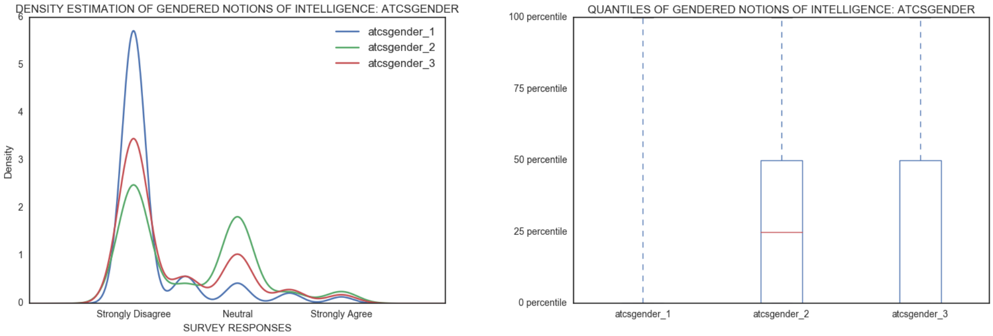
Figure 0.5: **Quantiles for dimension** `atct`. *Self-reported attitudes about computational thinking.*



- atcsgender_1: Women are less capable of success in CS than men.

- atcsgender_2: Women are smarter than men.

- atcsgender_3: Men have better math and science abilities than women.



(a)

(b)

Figure 0.6: **Dimensions of `atcsgender`.** *Figure (a) Density estimation for the dimension. Figure (b) Boxplot for the same dimension.*

ALGORITHMS AND TECHNIQUES

For the problem of determining the factors that predict intro CS experience based on gender, 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) I selected an 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) I selected a Support Vector Machine (SVMs) because they are very robust classifiers and *more importantly*, they have a method to correct for class imbalances.

(d) Finally I selected a Decision Tree classifier because it lends itself to interpretability. For this problem domain, it is not just satisfactory for me 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

Before I start selecting which classifier I want to proceed with, I need a **baseline** score on which I can evaluate the practical value of datamining for this problem. Since this project is applying machine learning to a novel dataset, I do not have standard benchmarks I can measure against. As such, I have decided to use a simple *majority* classifier which always selects the majority class of the training set.

IMPLEMENTATION

I implemented the four learning algorithms. For each of the learners I implemented the baseline algorithm using a stratified shuffle split cross validation with 10 folds and calculated the F_1 scores and looked at the confusion matrices respectively.

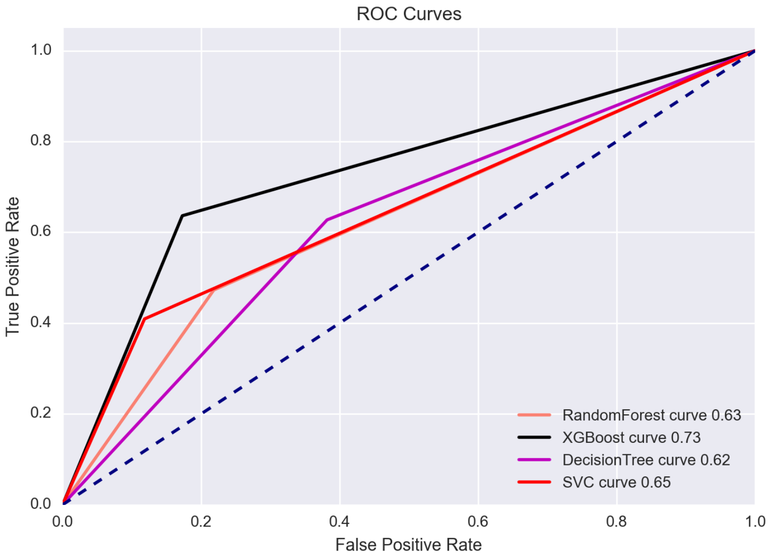
From running these baseline classifiers, I selected the xgboost classifier, 0.8c, as the right learner for this problem because it had the highest expected value score. On this problem, Random Forest classifier and the Support Vector Machine did not give a better perfor-

Table 0.1: Scores

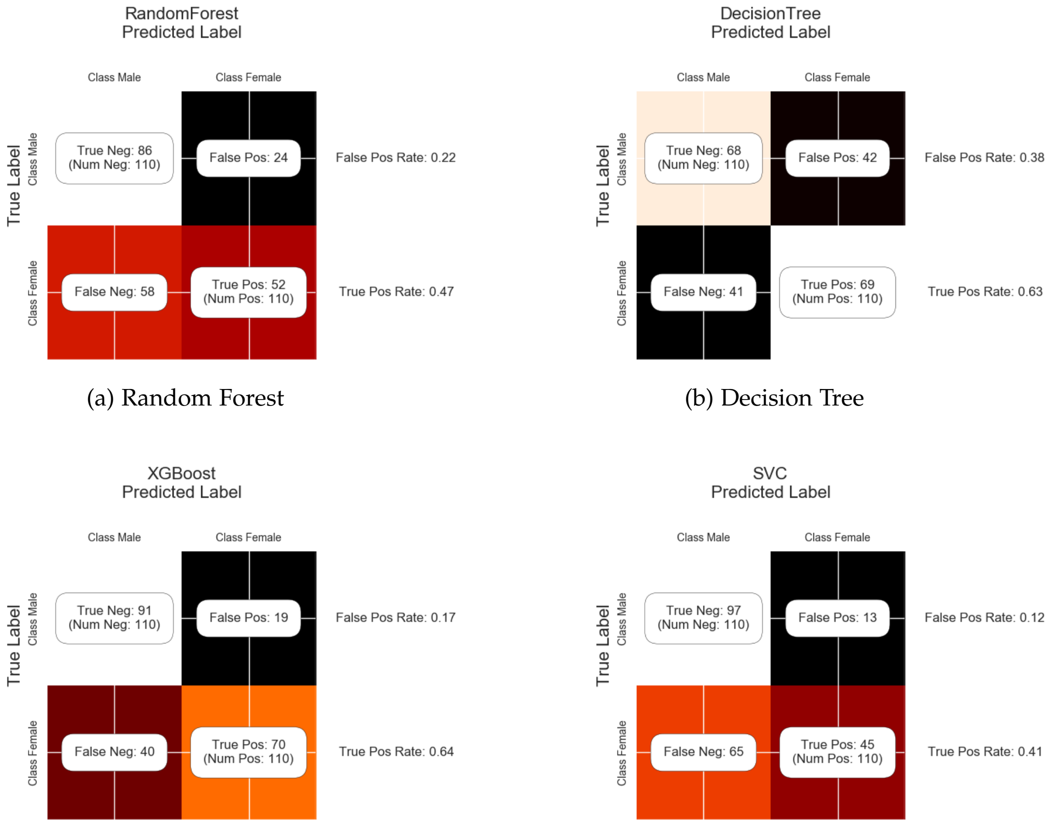
Result of training the baseline classifiers			
Classifier	Training Score	Prediction Score	Expected Value
Majority	-	58.01%	0.0
SVC	55.39%	53.57%	1.63%
DecisionTree	51.08%	62.44%	2.63%
RandomForestClassifier	54.93%	55.91%	1.87%
XGBClassifier	59.69%	70.35%	3.11%

mance than the *majority* classifier. While the Decision Tree did well, it was not as robust as the XGBoost classifier.

Figure 0.7: **ROC Curve.**



Its interesting to note that the Decision Tree classifier, had the highest true positive rate at 0.63, however, its false positive rate was a staggering 0.38! This means that it cannot find a meaningful sets of conditions for separating males from females in the dataset. The Support Vector Machine had the lowest false positive rate but did not beat the majority classifier because its true positive rate was abysmal. Where as, XGBoost does satisfactory on both fronts. You can really see this from looking at the ROC curves of the classifiers, 0.7.



(a) Random Forest

(b) Decision Tree

(c) XgBoost

(d) SVC

Figure 0.8: **Confusion Matrices of Baseline Classifiers**

RESULTS

MODEL EVALUATION AND VALIDATION

I am going to tune my model based on some heuristics about the kinds of value ranges that are suitable for the hyper-parameters I want to learn. I will be using these values ranges for the hyper-parameters:

- Parameters for Tree Booster
    - `colsample_bytree`
        - subsample ratio of columns when constructing each tree
        - Range (0, 1], default 0.6, tuned on $[0.4, 0.6, 0.8, 1.0]$
    - `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 $[0.01, 0.007, 0.005, 0.004, 0.003, 0.003, 0.003, 0.002, 0.002]$

I will implement the tuning using sklearn's `GridSearch` in conjunction with a {k=10 fold} `StratifiedShuffleSplit` function.

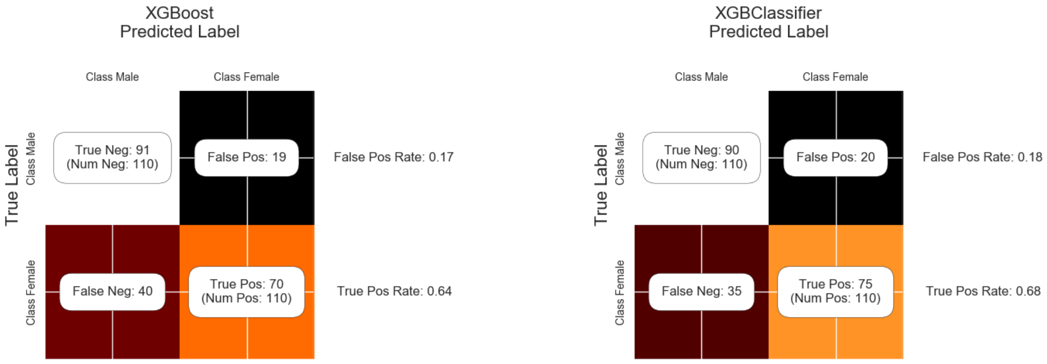
RESULTS OF TUNING

Once I performed the search through the hyper-parameter space to find the combination of hyper-parameters that maximized the performance of the selected classifier, I was able to **improve** the previous F_1 score by **2.82%**, to achieve a prediction score of 73.17%.

From figures 0.9a and 0.9b, one can see that the **false negative** count for the female class has gone from 40 down to 35. This decision cost us a very small increase in the **false positive** count of the male class from 19 to 20. This is not too bad, so I will stick with this improved model.

Here is the final model for classifying gender in introductory CS.

```
Best accuracy obtained: 73.17%
Parameters:
    n_estimators: 900
    subsample: 0.7
    learning_rate:  0.005
    colsample_bytree: 1.0
    max_depth: 6
```



(a) Base Model

(b) Tuned Model

Figure 0.9: **Confusion Matrix**

FEATURE IMPORTANCE

To identify factors that predict experience based on gender, I will then extract the top features responsible for discriminating the data and then expand the final step to:

- Explore the various parameters around feature splitting
    - Xgboost algorithm feature importance score
    - Information gain
    - Cover

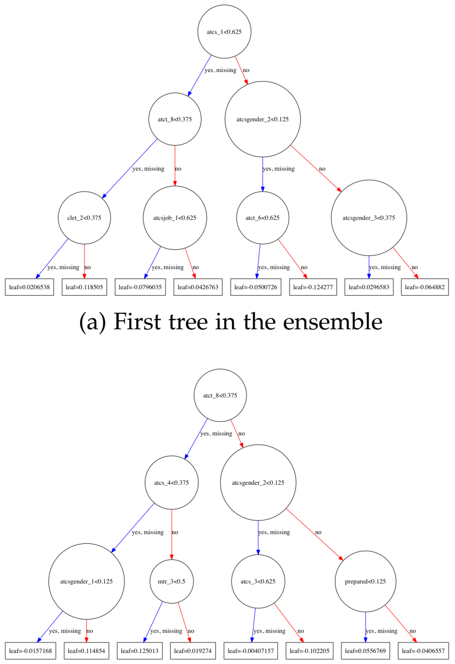
Information gain, lets us know how valuable a feature is in discriminating the dataset. That is, if we know the information gain of a feature, we can know how much it would contribute to the knowledge of the value of the target label. One can think on information gain as a measurement of **informativeness** of a feature with respect to the target class. While the 'cover' is the sum of second order gradient in each node, and intuitively it represents the number of data points affected by the split.

There are two things that need consideration when using xgBoost for understanding feature importance: the features that are doing the *most* work in splitting the data, and the automatically generated feature importance ranking that is done in by the xgBoost algorithm.

I plotted some estimators in the xgboost learner to see which features are doing the most work in splitting the data. I chose to focus on the **first** and **second** tree in the ensemble. On simple models, the first two trees may be enough to gain a *strong* understanding. I then compare the output generated by these trees to the features generated by the model's own feature selection algorithm.

Optimized Trees

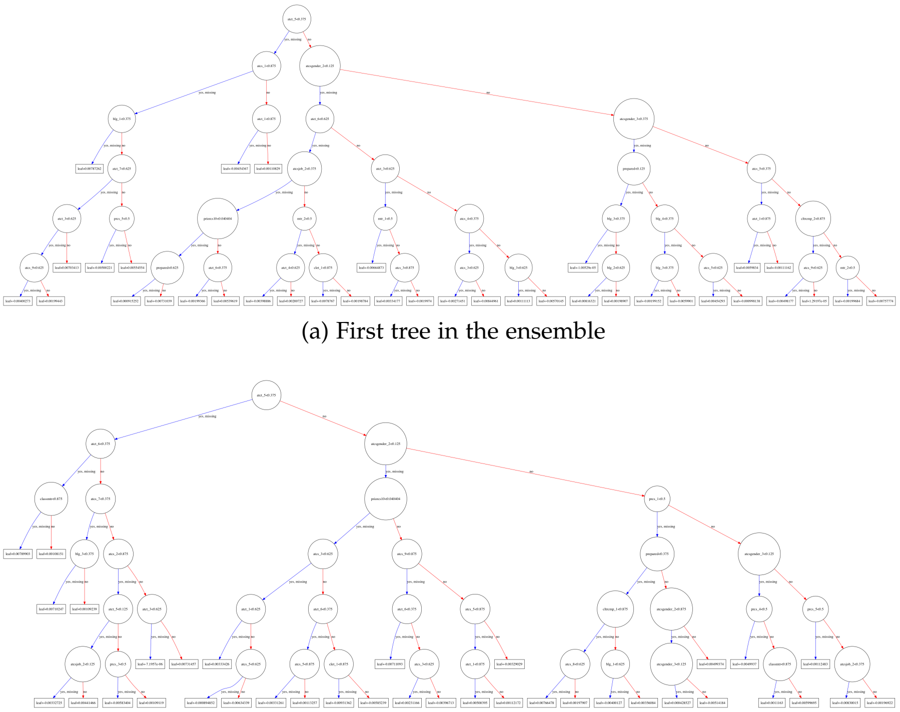
The tuned model has a more complex tree that goes down six levels, for each of its estimators. This model segmented the data into 36 distinct types; you can see this by counting the number of leaf nodes.



(a) First tree in the ensemble

(b) Second tree in the ensemble

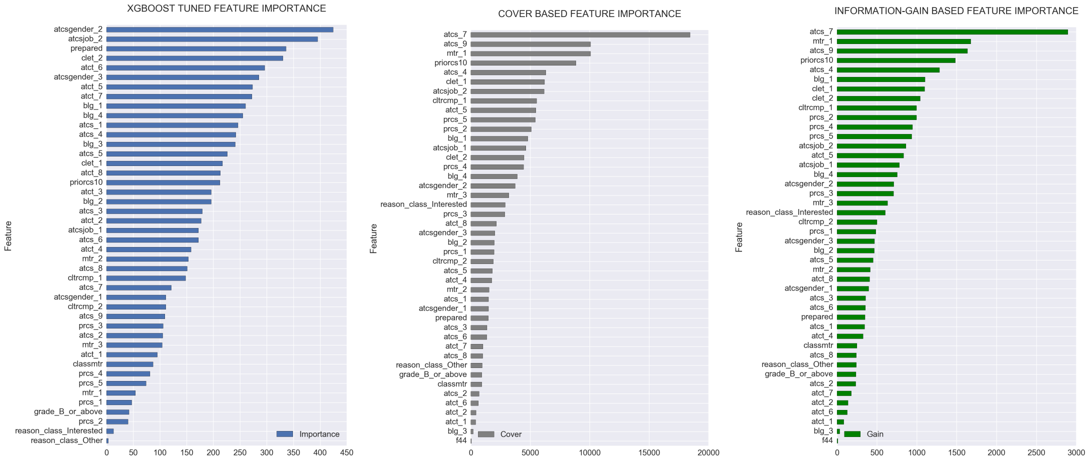
Figure 0.10: **XgBoost base model decision trees**



(a) First tree in the ensemble

(b) Second tree in the ensemble

Figure 0.11: **Tuned XgBoost estimator decision tree.**



(a) Feature Importance Tuned Model

(b) Feature Importance Based on Cover

(c) Feature Importance Based on Information Gain

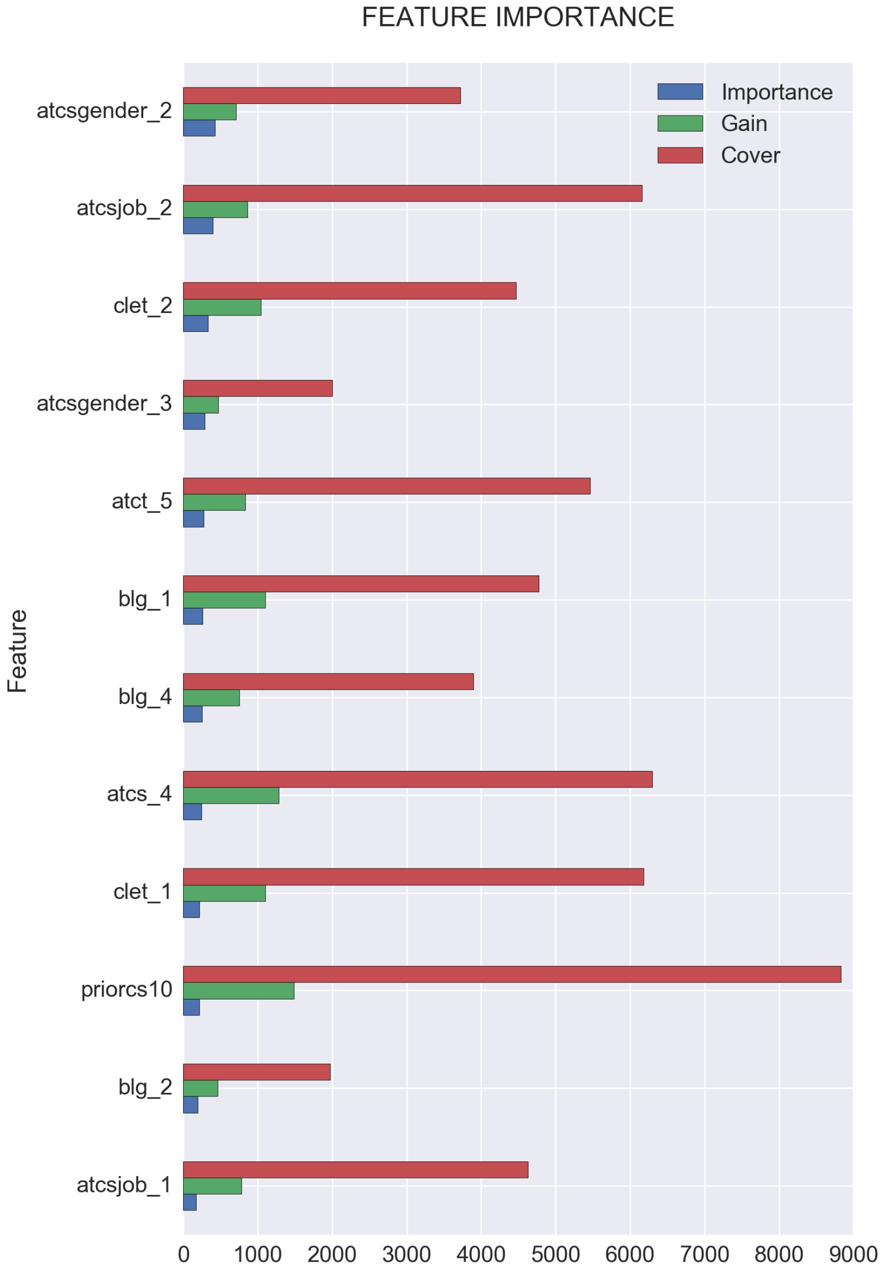
Figure 0.12: **XgBoost Feature Importance**

Table 0.2: Feature Importance

XGBoost Feature Importance		
Rank	Coded item	Description
1	atcsgender_2	Women are smarter than men.
2	atcsjobs_2	My career goals do not require that I learn computing skills.
3	clet_2	I think about the ethical, legal, and social implications of computing.
4	atct_6	I am good at building things.
5	prepared	How prepared did you feel about this class before it started?
6	atcsjob_1	Knowledge of computing will allow me to secure a good job.
7	clet_1	I work well in teams.
8	atct_7	I am good at ignoring irrelevant details to solve a problem.
9	blg_1	In this class, I feel I belong.
10	atct_3	I am good at using online search tools.

CONCLUSION

Figure 0.13: **Final Features.** *These features predict Intro CS Experience Based on Gender.*



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.
- I can learn to understand computing concepts.
- 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.

CAREER DRIVEN BELIEFS ABOUT COMPUTER SCIENCE

- Knowledge of computing will allow me to secure a good job.
- My career goals do not require that I learn computing skills.

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?