Census Income Modeling

Anthony Banks

12/29/2019

1.0 Introduction

The following report documents the methods and procedures I used to create prediction models on a dataset of census information. I obtained the dataset from Kaggle.com at the following URL: https://www.kaggle.com/uciml/adult-census-income. The original dataset consisted of the following 15 columns:

- age (integer) - The age of each entry
- workclass (string) - The workclass of each entry. Options include: private, local government, federal government, self-employed
- fnlwgt (integer) - The final weight of that row. This relates to how many unique census entries were of this row's characteristics. In order to preserve the tidy format of the dataset, I did not include the fnlwgt column in my computations.
- education (string) - The highest form of education for each entry. Options include: HS-grad, Masters, 10th
- education.num (integer) - The total number of years spent in education for each entry
- marital.status (string) - The marital status of each entry
- occupation (string) - The occupation of each entry
- relationship (string) - The family-relationship held by each entry. Options include: Mother, Father, Not-in-Family, Unmarried
- race (string) - The race of each entry
- sex (string) - The gender of each entry
- capital.gain (integer) - The capitol gain of each entry
- capital.loss (integer) - The capitol loss of each entry
- hours.per.week (integer) - The hours per week worked by each entry
- native.country (string) - The native country of each entry
- income (string) - The annual income of each entry as defined categorically as less than or equal to $50,000 (<=50K) and greater than $50,000 (>50K).

I used the census dataset to generate models that predict income status as defined by **less than or equal to $50,000** or **greater than $50,000** based on that entry's status for the other attributes shown above.

2.0 Methods

2.1 Data Preparation

I downloaded the dataset as a zipped csv file, read it into R, and examined it for dimensionality and missing values.

```
## The original dataset had 32561 rows,  15 columns, and 0 missing values.
```

Next, I randomly split the original dataset into training and testing datasets comprised of 90% and 10% of the original dataset, respectively. I used the training dataset to create visualizations, look for trends, and to ultimately train the models. The testing dataset was strictly used for the sole purpose of evaluating the final success of each model. The use of the testing dataset for anything beyond the evaluation of each final model could overtrain the model and provide poor success on future datasets.

```
## The training dataset had 29304 rows and 15 columns.
## The testing dataset had 3257 rows and 15 columns.
```

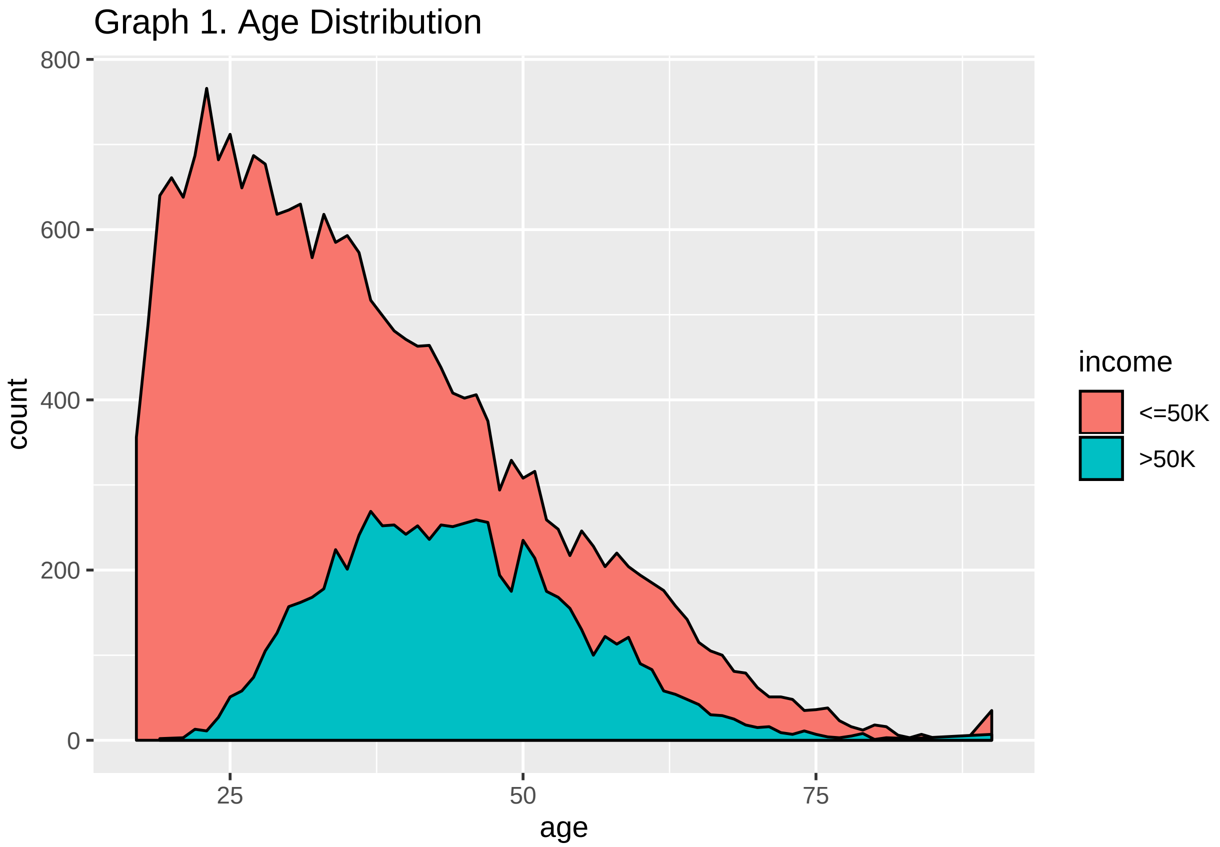
2.2 Data Visualization and Interpretation

I created a summary table and several graphs of the data so that I could explore the data for trends and outliers. The summary table below shows the distribution of the two income categories. The ">50K" income category was outnumbered approximately 3 to 1. The uneven distribution of the income categories is significant enough to categories the dataset as unbalanced.

Table 1: Income Status Distribution in the Training Dataset

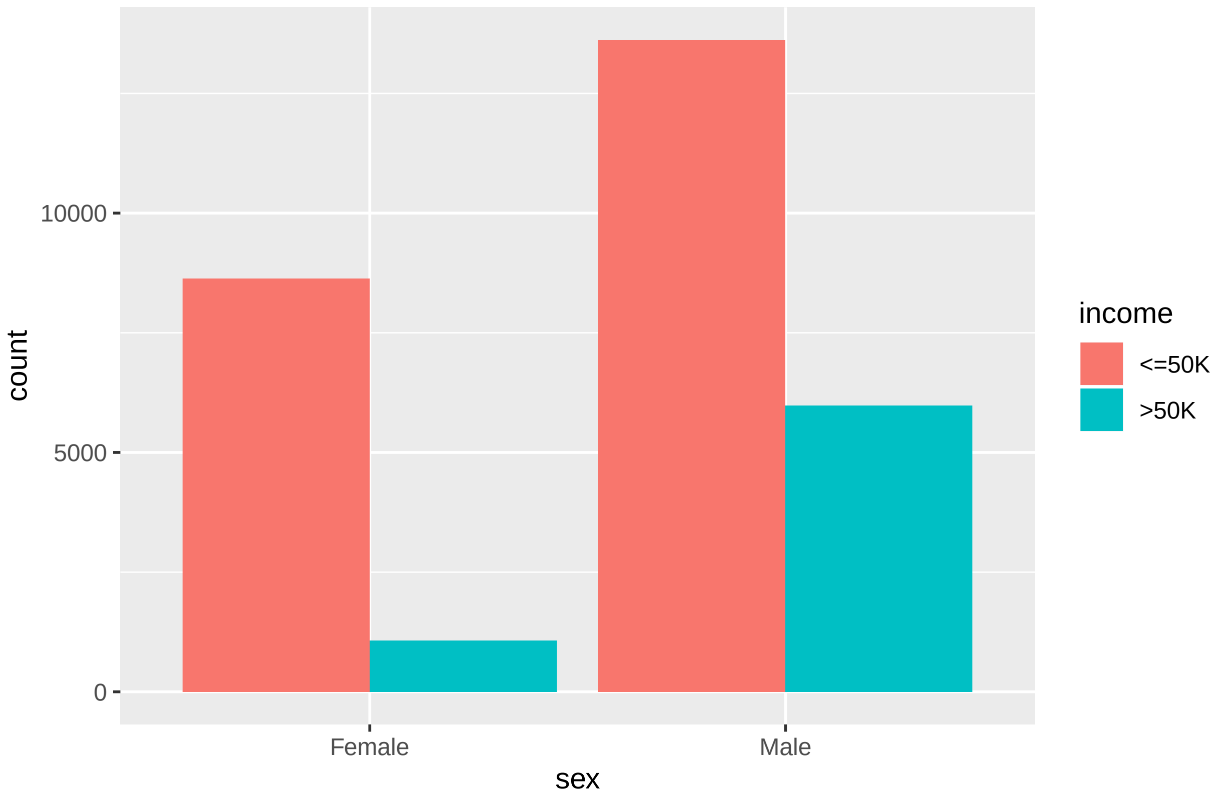
income	Count	Percent
<=50K	22248	75.9
>50K	7056	24.1

A majority of the dataset's columns are categorical, thus making scatter plots insignificant. I created histograms and density plots to visualize the distribution of the income statuses for the other variables in the dataset.

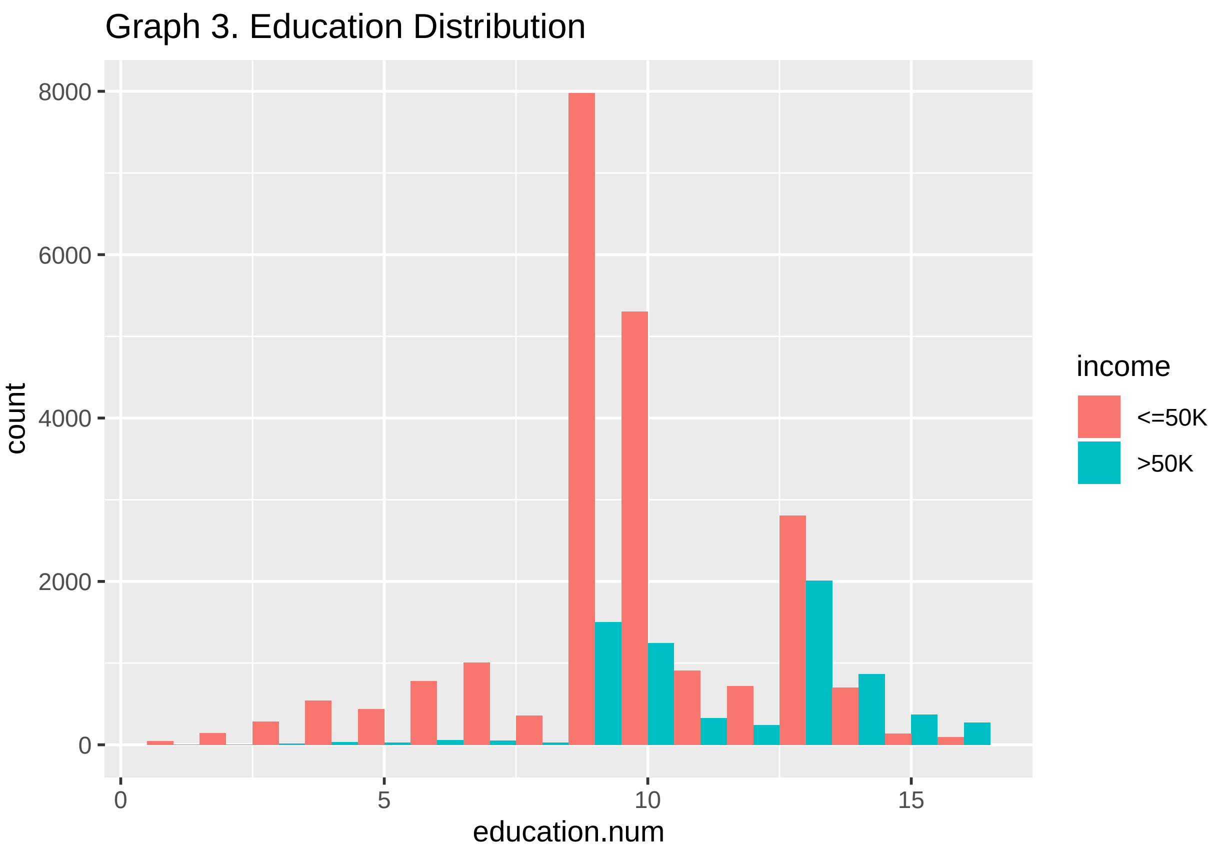


Graph 1 shows the two income categories have a significantly different trend with age. The most common age for the <=50K group was approximately 24 years old, and the group had a decrease in entries for each successive age. Alternatively, the most common age for the >50K group was in the range of approximately 30 to 50 years old.

Graph 2. Gender Distribution



Graph 2 reveals there were more male entries than female entries for both income categories, but males significantly outnumber females in the >50K group. Graph 2 indicates that gender could be a significant predictor of income category.



Graph 3 shows that a majority of the entries have 9, 10, or 13 years of education for both income categories. Those years of education relate to a high school education, one year of college, and 4 years of college, respectively. Graph 3 shows the >50K group had a higher average years of education.

Graph 4. Race Distribution

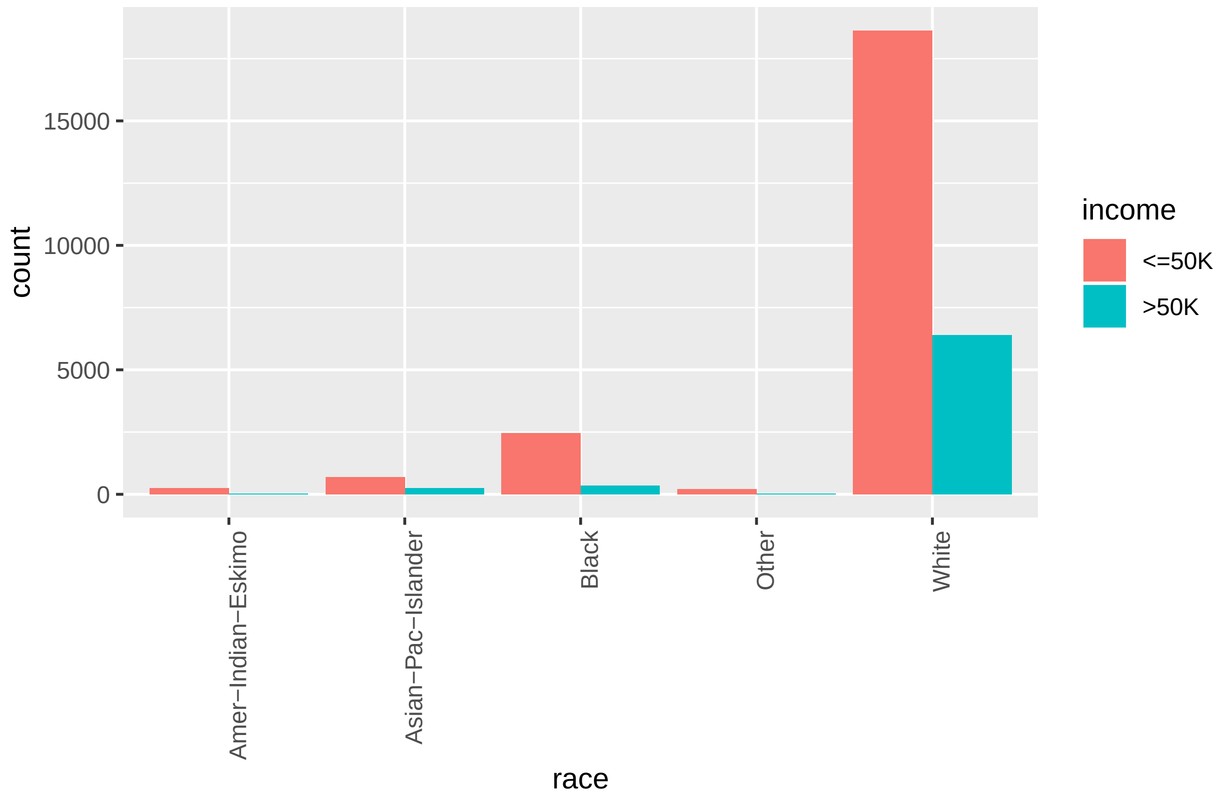
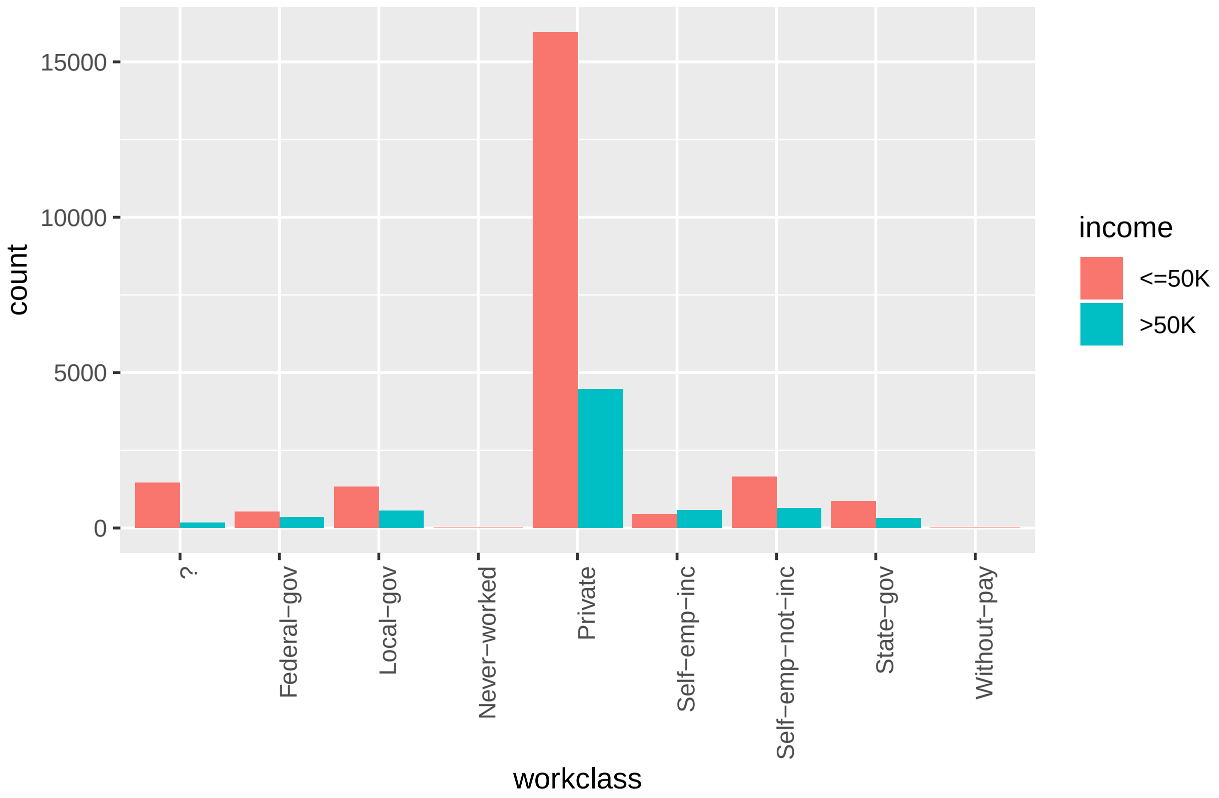


Table 2: Income Status Distribution in the Training Dataset by Race

race	Count <=50K	Count >50K	Percent >50K
Amer-Indian-Eskimo	258	33	11.340
Asian-Pac-Islander	683	247	26.559
Black	2453	356	12.674
Other	221	22	9.053
White	18633	6398	25.560

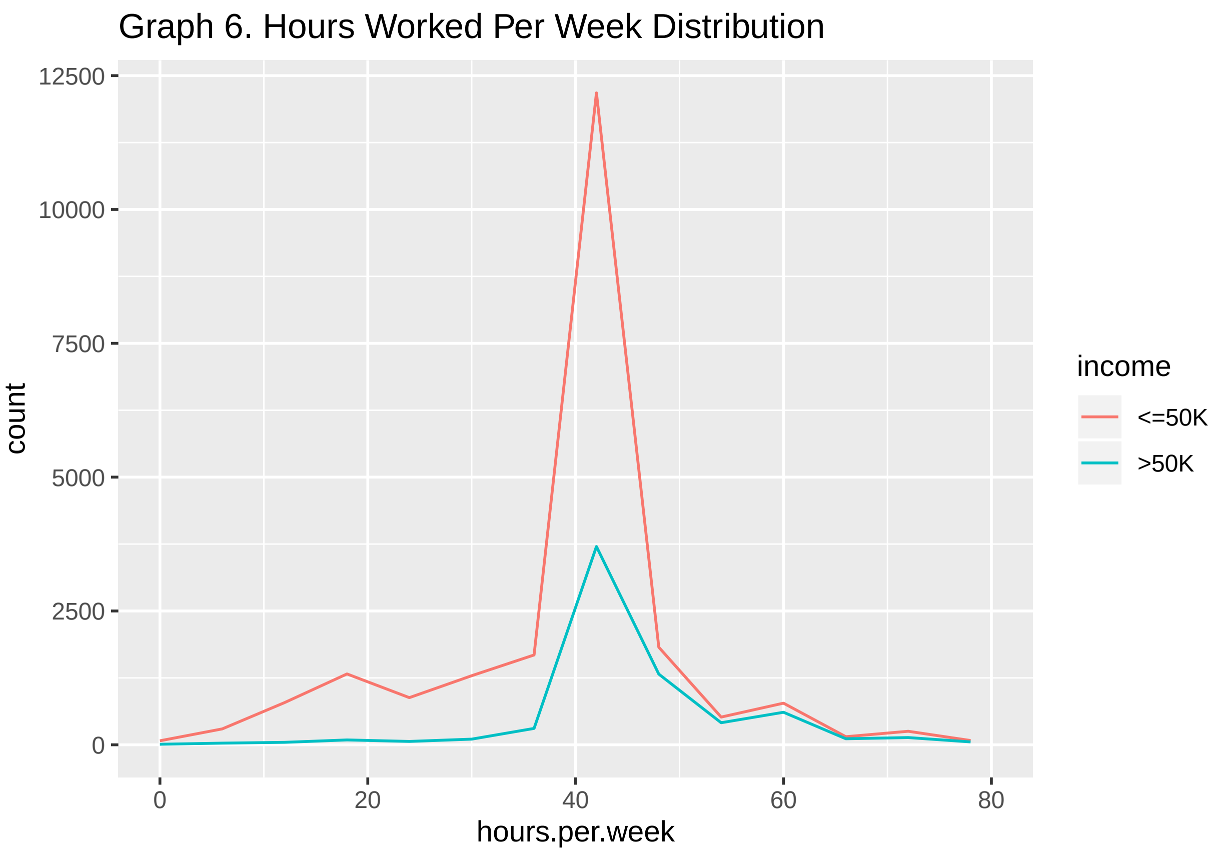
Graph 4 reveals that a majority of the entries are White, but it also shows that very few non-White entries were in the >50K income category. Table 2 shows that the percentage of Asian-Pac-Islander and White entries in the >50K category is more than double the percentage in the >50K income category for the other races.

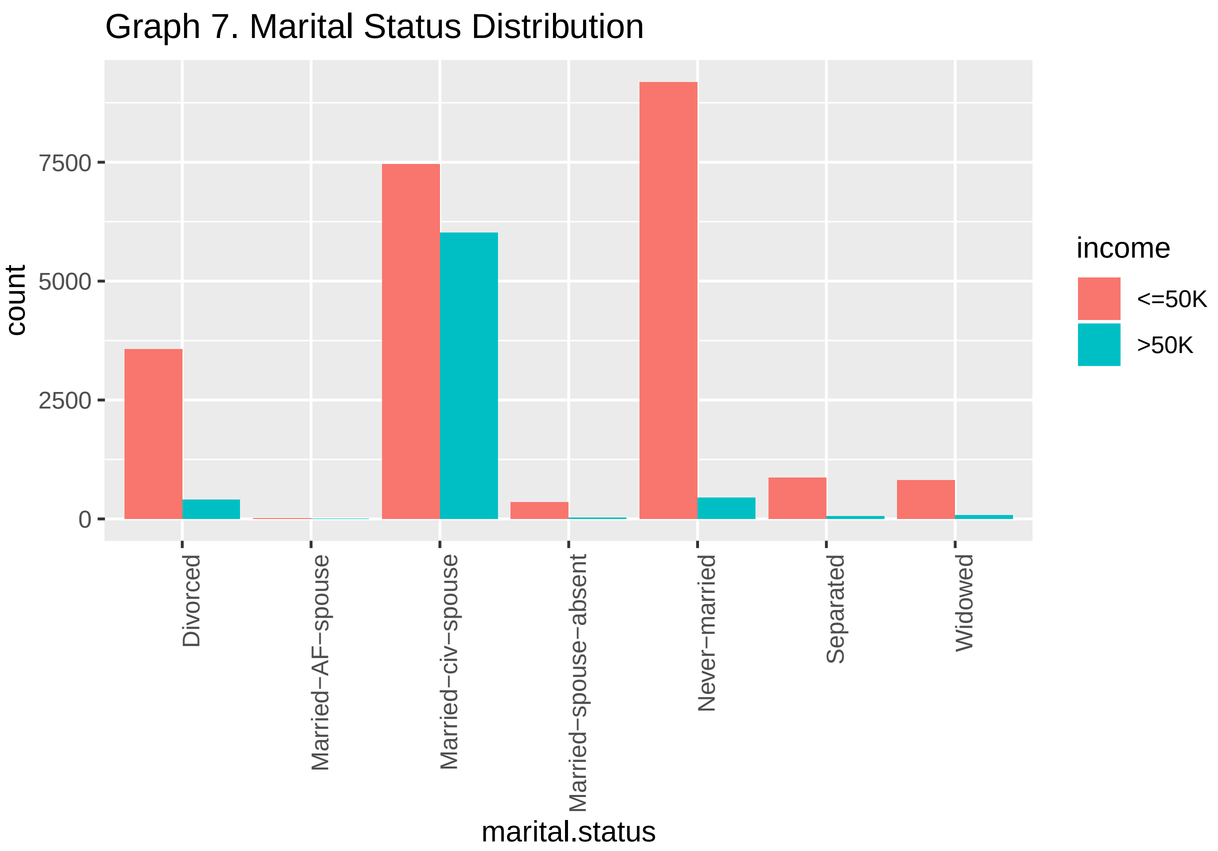
Graph 5. Workclass Distribution



Graph 5 shows the distribution of workclass among the entries. A very large majority of the entries were in the private workclass, and the non-private workclasses, except for “?”, “Never-worked”, and “Without-pay”, had a higher percentage of >50k income category entries than observed in the private workclass.

Graph 6 shows that both income categories were predominantly in the range of 38 to 44 hours per week, but the >50K income category has a higher percentage of entries who work more than 44 hours per week.


Graph 6. Hours Worked Per Week Distribution
12500
10000
7500
5000
2500
0
count
0
20
40
60
80
hours.per.week
income
<=50K
>50K



Graph 7 shows that a very high majority of the >50K income category are in the "Married-civ-spouse" marital status category. It appears that the other marital status options could be used to predict an entry as not being in the >50K income group.

2.3 Modelling

I trained each model using the training dataset described above, modified such that the data for "fnlwgt", "relationship", and "education" were removed; those columns were removed because "fnlwgt" does not apply for the purposed of evaluating the dataset as if each row is for one unique entry. Furthermore, "relationship" and "education" were reductant information that could be deduced from marital status and sex, and "education.num", respectively.

I used the testing dataset to evaluate the permformance of each model based on it's F1-score, accuracy, sensitivity, and specificity; but ranked each model's success based on their F1-score.

2.3.1 Model 1

For Model 1, I predicted <=50K for every entry. Although I knew this model would be flawed, it provided useful baseline results to compare future models against. Model 1 was reported with a sensitivity of 1.00 and a specificity of 0.00, because it was correct 100% of the time when the true value was <=50K, and it was correct 0% of the time when the true value was >50k. Because the complete dataset is weighted in favor of the <=50K income category, Model 1 was reported with suprisingly high F1-score and accuracy.

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction <=50K >50K
##      <=50K  2472  785
##      >50K      0    0
##
##                Accuracy : 0.759
##                  95% CI : (0.7439, 0.7736)
##     No Information Rate : 0.759
##     P-Value [Acc > NIR] : 0.5096
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000
##             Specificity : 0.000
##          Pos Pred Value : 0.759
##          Neg Pred Value :   NaN
##              Prevalence : 0.759
##          Detection Rate : 0.759
##    Detection Prevalence : 1.000
##       Balanced Accuracy : 0.500
##
##        'Positive' Class : <=50K
##
```

```
##                F1  Accuracy Sensitivity Specificity
## Model 1 0.8629778 0.7589807           1           0
```

2.3.2 Model 2

For Model 2, I trained a linear regression model. There are no training parameters for the linear regression model, so it could not be fine tuned. The reported F1 score, accuracy, and specificity of Model 2 increased from Model 1, and the sensitivity decreased. However, the reported specificity of Model 2 is still significantly low. Model 2 incorrectly predicts <=50K on approximately half the entries in the >50K group.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  2329  407
##      >50K    143  378
##
##                Accuracy : 0.8311
##                  95% CI : (0.8178, 0.8439)
##     No Information Rate : 0.759
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4786
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9422
##             Specificity : 0.4815
##          Pos Pred Value : 0.8512
##          Neg Pred Value : 0.7255
##              Prevalence : 0.7590
##          Detection Rate : 0.7151
##    Detection Prevalence : 0.8400
##       Balanced Accuracy : 0.7118
##
##        'Positive' Class : <=50K
##
```

```
##                F1  Accuracy Sensitivity Specificity
## Model 1 0.8629778 0.7589807   1.0000000   0.0000000
## Model 2 0.8943932 0.8311329   0.9421521   0.4815287
```

2.3.3 Model 3

For Model 3, I trained a generalized linear model (GLM). Similar to the linear regression model, GLM does not have any training parameters. The reported F1-score, accuracy, and specificity of Model 3 increased from Model 2, and the sensitivity decreased. The specificity of Model 3 is still lower than ideal, but it increased significantly from Model 2.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  2314  334
##      >50K    158  451
##
##                Accuracy : 0.8489
##                  95% CI : (0.8362, 0.8611)
##     No Information Rate : 0.759
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5529
##
##  Mcnemar's Test P-Value : 3.031e-15
##
##             Sensitivity : 0.9361
##             Specificity : 0.5745
##          Pos Pred Value : 0.8739
##          Neg Pred Value : 0.7406
##              Prevalence : 0.7590
##          Detection Rate : 0.7105
##    Detection Prevalence : 0.8130
##       Balanced Accuracy : 0.7553
##
##        'Positive' Class : <=50K
##
```

```
##                F1  Accuracy Sensitivity Specificity
## Model 1 0.8629778 0.7589807   1.0000000   0.0000000
## Model 2 0.8943932 0.8311329   0.9421521   0.4815287
## Model 3 0.9039063 0.8489407   0.9360841   0.5745223
```

2.3.4 Model 4A and Model 4B

For Model 4A and Model 4B, I trained regression tree models. Regression tree models have a training parameter, the complexity parameter. I trained Model 4A without adjusting the range of complexity parameter values, and trained Model 4B with a wide range of complexity parameter values.

The overall performance of Model 4A was worse than Model 3.

Model 4B had overall performance similar to Model 3. Model 4B's F1-score and sensitivity were greater than those values for Model 3, but Model 3's accuracy and specificity were greater than those value for Model 4B.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  2347  406
##      >50K    125  379
##
##                Accuracy : 0.837
##                  95% CI : (0.8238, 0.8495)
##     No Information Rate : 0.759
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4924
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9494
##             Specificity : 0.4828
##          Pos Pred Value : 0.8525
##          Neg Pred Value : 0.7520
##              Prevalence : 0.7590
##          Detection Rate : 0.7206
##    Detection Prevalence : 0.8453
##       Balanced Accuracy : 0.7161
##
##        'Positive' Class : <=50K
##
```

```
##                 F1  Accuracy Sensitivity Specificity
## Model 1  0.8629778 0.7589807   1.0000000   0.0000000
## Model 2  0.8943932 0.8311329   0.9421521   0.4815287
## Model 3  0.9039063 0.8489407   0.9360841   0.5745223
## Model 4A 0.8983732 0.8369665   0.9494337   0.4828025
```

Graph 8. Estimated Accuracy vs Complexity Parameter in Model 4B

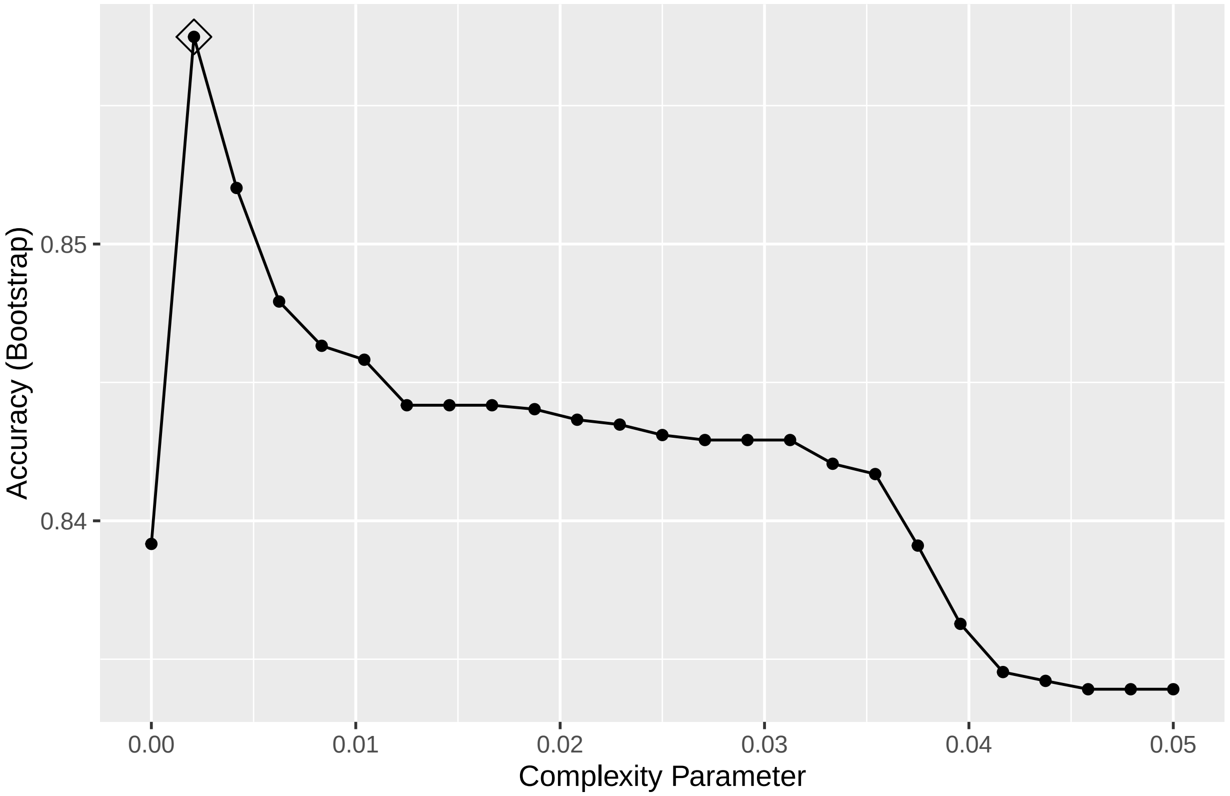
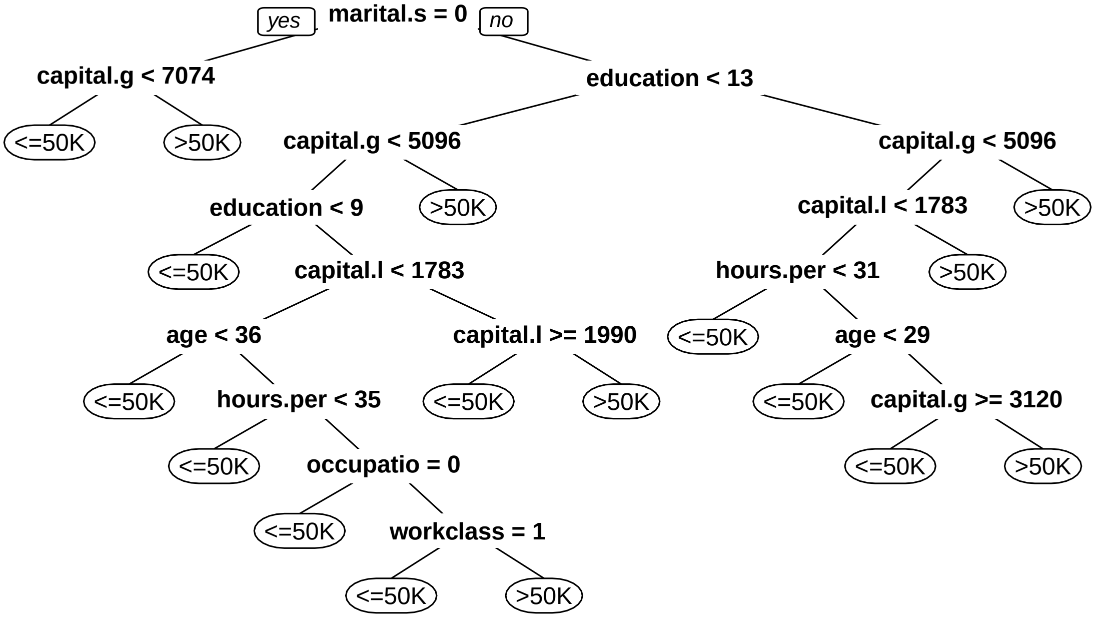


Figure 1. Decision Tree of Model 4B



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction <=50K >50K
##      <=50K  2348  369
##      >50K    124  416
##
##                Accuracy : 0.8486
##                  95% CI : (0.8359, 0.8608)
##     No Information Rate : 0.759
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.537
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9498
##             Specificity : 0.5299
##          Pos Pred Value : 0.8642
##          Neg Pred Value : 0.7704
##              Prevalence : 0.7590
##          Detection Rate : 0.7209
##    Detection Prevalence : 0.8342
##       Balanced Accuracy : 0.7399
##
##        'Positive' Class : <=50K
```

```
##
```

```
##                 F1  Accuracy Sensitivity Specificity
## Model 1  0.8629778 0.7589807   1.0000000   0.0000000
## Model 2  0.8943932 0.8311329   0.9421521   0.4815287
## Model 3  0.9039063 0.8489407   0.9360841   0.5745223
## Model 4A 0.8983732 0.8369665   0.9494337   0.4828025
## Model 4B 0.9049913 0.8486337   0.9498382   0.5299363
```

3.0 Results

Model 3, GLM, and Model 4B, regression tree model with an adjusted complexity parameter range, produced similar model success, however, Model 4B resulted in the highest F1-score and therefore was selected as the best model in this report. All of the models showed significant bias towards predicting <=50K because of the unbalanced dataset.

```
##                 F1  Accuracy Sensitivity Specificity
## Model 1  0.8629778 0.7589807   1.0000000   0.0000000
## Model 2  0.8943932 0.8311329   0.9421521   0.4815287
## Model 3  0.9039063 0.8489407   0.9360841   0.5745223
## Model 4A 0.8983732 0.8369665   0.9494337   0.4828025
## Model 4B 0.9049913 0.8486337   0.9498382   0.5299363
```

4.0 Conclusions

I used the census dataset to generate models that predict income status as defined by **less than or equal to $50,000** or **greater than $50,000** based on that entry's status for the other attributes in the dataset. I had the most success, as evaluated by the F1-score, when using a regression tree model with an adjusted range of the complexity parameter. However, all of the models were significantly biased from the unbalanced nature of the dataset. In order to improve the success of predicting income status based on the parameters presented in the original dataset, future modelling efforts should include ways to counteract the unbalanced nature of the dataset, such as with undersampling or oversampling.