Census Data Final Project

Harvard EdX Professional Data Science Certificate

Samara Angel

Spring 2021

Contents

Introduction

Data analysis, data visualization, and machine learning prediction problems can all prove extremely important to understanding inequities that are at play in our society. In this project, I look at the US Adult Census dataset, extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker. Their data is available both on Kaggle, and through the UC Irvine (UCI) Machine Learning Repository. This data set was developed by Kohavi and Becker for machine learning challenges – specifically the challenge of developing predictor models to determine if an individuals income is greater than 50,000 (>50k) or less than/equal to 50,000 (<=50k). My goal in this project is to complete that predictor challenge. As stated alongside the data about its creation, "a set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1) && (HRSWK>0))." Prior to data cleaning, the data set contained 32561 rows and 15 columns. The initial columns were as follows (as seen in the UCI data set summary):

- **age**: continuous.
- **workclass**: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- **fnlwgt**: continuous.
- **education**: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- **education-num**: continuous.
- **marital-status**: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- **occupation**: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- **relationship**: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- **race**: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- **sex**: Female, Male.
- **capital-gain**: continuous.
- **capital-loss**: continuous.
- **hours-per-week**: continuous.
- **native-country**: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

I begin by cleaning and visualizing the data. Specifically, I remove the workclass, native_country, relationship, capital_gain, capital_loss, and hours_per_week categories, as I won't be using those in my analysis. I then complete one-hot encoding and dummy encoding of variables, such that I can create two datasets: one that contains numeric variables and one that contains both factor variables and numeric variables (see the Generalized Linear Models section for an explanation of one-hot and dummy encoding). I do this for the ease of the models I later run. For the factor data and for the numeric data, I use an identical index to partition both data sets into a test set, containing 20% of the data, and a training set, containing 80% of the data. I will use these data in developing my predictor models.

I use three metrics throughout to evaluate the predictor models I develop for this project: overall accuracy, sensitivity, and specificity. Using definitions based on those of Professor Irizarry, I define overall accuracy as the proportion overall that has been predicted correctly by a given model. However, simply using overall accuracy as a metric can be misleading. Prevalence impacts this overall accuracy metric. For instance, as I will later show, for individuals with income >50k, there is a prevalence of 0.7483, meaning that the prevalence of individuals with income <=50k is 0.2517. Upon inspection of the data, 90% of individuals in the data set are from the United States, 8% are from Mexico, and 2% from other countries. In 1994,

the median income of households in the United States was 32,264 dollars. This implies that there would be theoretically a much higher prevalence of individuals with income <=50k than appear in our data. Because the data set is not balanced (a 50/50 split), this can impact the results. Having this bias in the training data can warp the meaning of overall accuracy. To solve the problem of using solely the overall accuracy metric, I will also be looking at sensitivity and specificity, which are defined for a binary outcome (such as income). For sensitivity and specificity, following definitions based on those of Professor Irizarry, I refer to positive outcomes as $Y = 1$ and negative outcomes as $Y = 0$. Sensitivity is an algorithms ability to predict a positive outcome when the actual outcome is positive ($\hat{Y} = 1$ when $Y = 1$); specificity is an algorithms ability to NOT predict a positive when the actual outcome is not a positive ($\hat{Y} = 0$ when $Y = 0$). Sensitivity and specificity must be used together in order to be meaningful; if not, for example, an algorithm could have perfect sensitivity by simply predicting a positive each time. These measures (overall accuracy, sensitivity, and specificity) are all accessible using the confusionMatrix function in R.

Link to the UCI Machine Learning US Adult Census Data Page: https://archive.ics.uci.edu/ml/datasets/Census+Income

Link to the Kaggle US Adult Census Data Page: https://www.kaggle.com/uciml/adult-census-income?select=adult.csv

Link to the Data.World US Adult Census Data Page: https://data.world/uci/census-income/workspace/file?filename=adult.data.csv

Methods and Analysis

Data Cleaning and Pre-Processing

I begin by importing the US Adult Census data from data.world.

```
basic <- read.table("https://query.data.world/s/ofxhkosdcjihnprr3lctlxwzw44645")
```

I then attach the column names to the data set

```
colnames(basic) <- c("age", "workclass", "fnlwgt", "education",
    "education_num", "marital_status", "occupation", "relationship",
    "race", "sex", "capital_gain", "capital_loss", "hours_per_week",
    "native_country", "income")
```

Each entry in the data set currently ends with a comma, so I remove that comma.

```
data <- basic %>%
    mutate(age = gsub(",", "", basic$age), workclass = gsub(",",
        "", basic$workclass), fnlwgt = gsub(",", "", basic$fnlwgt),
        education = gsub(",", "", basic$education), education_dummy = gsub(",",
            "", basic$education_num), marital_status = gsub(",",
            "", basic$marital_status), occupation = gsub(",",
            "", basic$occupation), relationship = gsub(",", "",
            basic$relationship), race = gsub(",", "", basic$race),
        sex = gsub(",", "", basic$sex), capital_gain = gsub(",",
            "", basic$capital_gain), capital_loss = gsub(",",
            "", basic$capital_loss), hours_per_week = gsub(",",
            "", basic$hours_per_week), native_country = gsub(",",
            "", basic$native_country))
```

For simplicity, I then delete any rows with missing data, demarcated in this data set by "?".

```
dataset <- subset(data, data$age != "?" & data$workclass != "?" &
    data$fnlwgt != "?" & data$education != "?" & data$education_num !=
    "?" & data$marital_status != "?" & data$occupation != "?" &
    data$relationship != "?" & data$race != "?" & data$sex !=
    "?" & data$capital_gain != "?" & data$capital_loss != "?" &
    data$native_country != "?" & data$income != "?")
```

To make sure education level isn't impacted by an age restriction, I restrict age to higher than 18.

```
prep_set <- dataset %>%
    filter(age >= 18)
```

Here I take away variables that aren't relevant to my analysis. To make the data function the way I need it to for the generalized linear model test, I also reclassify the remaining variables as factors.

```
prep_set_2 <- prep_set %>%
    mutate(age = as.factor(age), fnlwgt = as.numeric(fnlwgt),
        race = as.factor(race), sex = as.factor(sex), education_dummy = as.factor(education_dummy),
        marital_status = as.factor(marital_status), occupation = as.factor(occupation),
        hours_per_week = as.factor(hours_per_week), income = as.factor(income)) %>%
    select(-workclass, -native_country, -relationship, -capital_gain,
        -capital_loss, -relationship, -education_num, -hours_per_week) %>%
    mutate(education_dummy = ordered(education_dummy, levels = c(1,
        2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)))
```

Here, I complete one-hot encoding using the dummyVars function to recode variables as numeric without ranking/order.

```
dummy <- dummyVars(" ~sex", data = prep_set_2)
prep_set_3 <- prep_set_2 %>%
    mutate(data.frame(predict(dummy, newdata = prep_set_2)))

dummy <- dummyVars(" ~marital_status", data = prep_set_3)
prep_set_4 <- prep_set_3 %>%
    mutate(data.frame(predict(dummy, newdata = prep_set_3)))

dummy <- dummyVars(" ~occupation", data = prep_set_4)
prep_set_5 <- prep_set_4 %>%
    mutate(data.frame(predict(dummy, newdata = prep_set_4)))

dummy <- dummyVars(" ~race", data = prep_set_5)
prep_set_6 <- prep_set_5 %>%
    mutate(data.frame(predict(dummy, newdata = prep_set_5)))

prep_set_factor <- prep_set_6 %>%
    mutate(income_dummy = ifelse(.$income == "<=50K", 0, 1))
```

Next, I recode variables as numeric for use in other models. I remove the factor-coded variables.

```
prep_set_numeric <- prep_set_factor %>%
    mutate(age = as.numeric(.$age), education_dummy = as.numeric(education_dummy)) %>%
    select(-marital_status) %>%
    select(-occupation) %>%
    select(-race) %>%
    select(-sex) %>%
    select(-income) %>%
    select(-education)
```

I then split both data sets into a training set and a final test set. I do this twice, once for the numeric and once for the factor data. The test sets will be 20% of two prep_sets I made above, and the train sets will be 80%. I chose those numbers because the data set is not extremely large and therefore benefits from an 80/20 split. I use income to standardize analysis. I do this twice, once for the factor-based prep_set_factor, and once for the numeric-based prep_set_numeric.

Here I partition the factor set.

```
set.seed(1)
test_index <- createDataPartition(y = prep_set_factor$income,
    times = 1, p = 0.2, list = FALSE)
train_factor <- prep_set_factor[-test_index, ]
test_factor <- prep_set_factor[test_index, ]
```

And here I partition the numeric set using the same index as for the factor set.

```
set.seed(1)
train_numeric <- prep_set_numeric[-test_index, ]
test_numeric <- prep_set_numeric[test_index, ]
```

Although separately partitioned and coded as factors vs. numeric variables, the two sets are identical in what their variables demonstrate.

Data Exploration and Visualization

In this section, I will explore the data set and visualize the data. I first look at the head of the train and test sets for both the factor and numeric data sets. Because the data have many columns, I truncate them to show only the first 9-10 columns for each set.

Table 1: Head of Train Factor Data Set

	age	fnlwgt	education	marital_status	occupation	race	sex	income	education_dummy	sex.Female
1	39	77516	Bachelors	Never-married	Adm-clerical	White	Male	<=50K	13	0
2	50	83311	Bachelors	Married-civ-spouse	Exec-managerial	White	Male	<=50K	13	0
3	38	215646	HS-grad	Divorced	Handlers-cleaners	White	Male	<=50K	9	0
4	53	234721	11th	Married-civ-spouse	Handlers-cleaners	Black	Male	<=50K	7	0
5	28	338409	Bachelors	Married-civ-spouse	Prof-specialty	Black	Female	<=50K	13	1
6	37	284582	Masters	Married-civ-spouse	Exec-managerial	White	Female	<=50K	14	1
7	49	160187	9th	Married-spouse-absent	Other-service	Black	Female	<=50K	5	1
8	52	209642	HS-grad	Married-civ-spouse	Exec-managerial	White	Male	>50K	9	0
11	37	280464	Some-college	Married-civ-spouse	Exec-managerial	Black	Male	>50K	10	0
13	23	122272	Bachelors	Never-married	Adm-clerical	White	Female	<=50K	13	1

Table 2: Head of Test Factor Data Set

	age	fnlwgt	education	marital_status	occupation	race	sex	income	education_dummy	sex.Female
9	31	45781	Masters	Never-married	Prof-specialty	White	Female	>50K	14	1
10	42	159449	Bachelors	Married-civ-spouse	Exec-managerial	White	Male	>50K	13	0
12	30	141297	Bachelors	Married-civ-spouse	Prof-specialty	Asian-Pac-Islander	Male	>50K	13	0
20	40	193524	Doctorate	Married-civ-spouse	Prof-specialty	White	Male	>50K	16	0
21	54	302146	HS-grad	Separated	Other-service	Black	Female	<=50K	9	1
22	35	76845	9th	Married-civ-spouse	Farming-fishing	Black	Male	<=50K	5	0
25	56	216851	Bachelors	Married-civ-spouse	Tech-support	White	Male	>50K	13	0
37	48	265477	Assoc-acdm	Married-civ-spouse	Prof-specialty	White	Male	<=50K	12	0
43	57	337895	Bachelors	Married-civ-spouse	Prof-specialty	Black	Male	>50K	13	0
47	29	271466	Assoc-voc	Never-married	Prof-specialty	White	Male	<=50K	11	0

Table 3: Head of Train Numeric Data Set

	age	fnlwgt	education_dummy	sex.Female	sex.Male	marital_status.Divorced	marital_status.Married.AF.spouse	marital_status.Married.civ.spouse	marital_status.Married.spouse.absent
1	22	77516	13	0	1	0	0	0	0
2	33	83311	13	0	1	0	0	1	0
3	21	215646	9	0	1	1	0	0	0
4	36	234721	7	0	1	0	0	1	0
5	11	338409	13	1	0	0	0	1	0
6	20	284582	14	1	0	0	0	1	0
7	32	160187	5	1	0	0	0	0	1
8	35	209642	9	0	1	0	0	1	0
11	20	280464	10	0	1	0	0	1	0
13	6	122272	13	1	0	0	0	0	0

Table 4: Head of Test Numeric Data Set

	age	fnlwgt	education_dummy	sex.Female	sex.Male	marital_status.Divorced	marital_status.Married.AF.spouse	marital_status.Married.civ.spouse	marital_status.Married.spouse.absent
9	14	45781	14	1	0	0	0	0	0
10	25	159449	13	0	1	0	0	1	0
12	13	141297	13	0	1	0	0	1	0
20	23	193524	16	0	1	0	0	1	0
21	37	302146	9	1	0	0	0	0	0
22	18	76845	5	0	1	0	0	1	0
25	39	216851	13	0	1	0	0	1	0
37	31	265477	12	0	1	0	0	1	0
43	40	337895	13	0	1	0	0	1	0
47	12	271466	11	0	1	0	0	0	0

I now need to make sure that the number of rows is identical between the numeric and factor data sets so that the two can be used interchangeably. I therefore examine the number of rows and columns in each data set. The difference between the train_numeric vs. train_factor and test_numeric vs. test_factor simply is that the numeric train and test set are missing the factor variables (that have not been one-hot encoded). The factor sets therefore have more columns, but both sets have the same number of rows. The two sets can therefore be used interchangeably depending on if a factor or numeric set is required for a given model.

Table 5: Number of Rows and Columns in the Train Numeric and Train Factor Data Sets

	Number of Rows	Number of Columns
Numeric	23866	32
Factor	23866	38

I now move to examining the income variable, the outcome variable for this project.

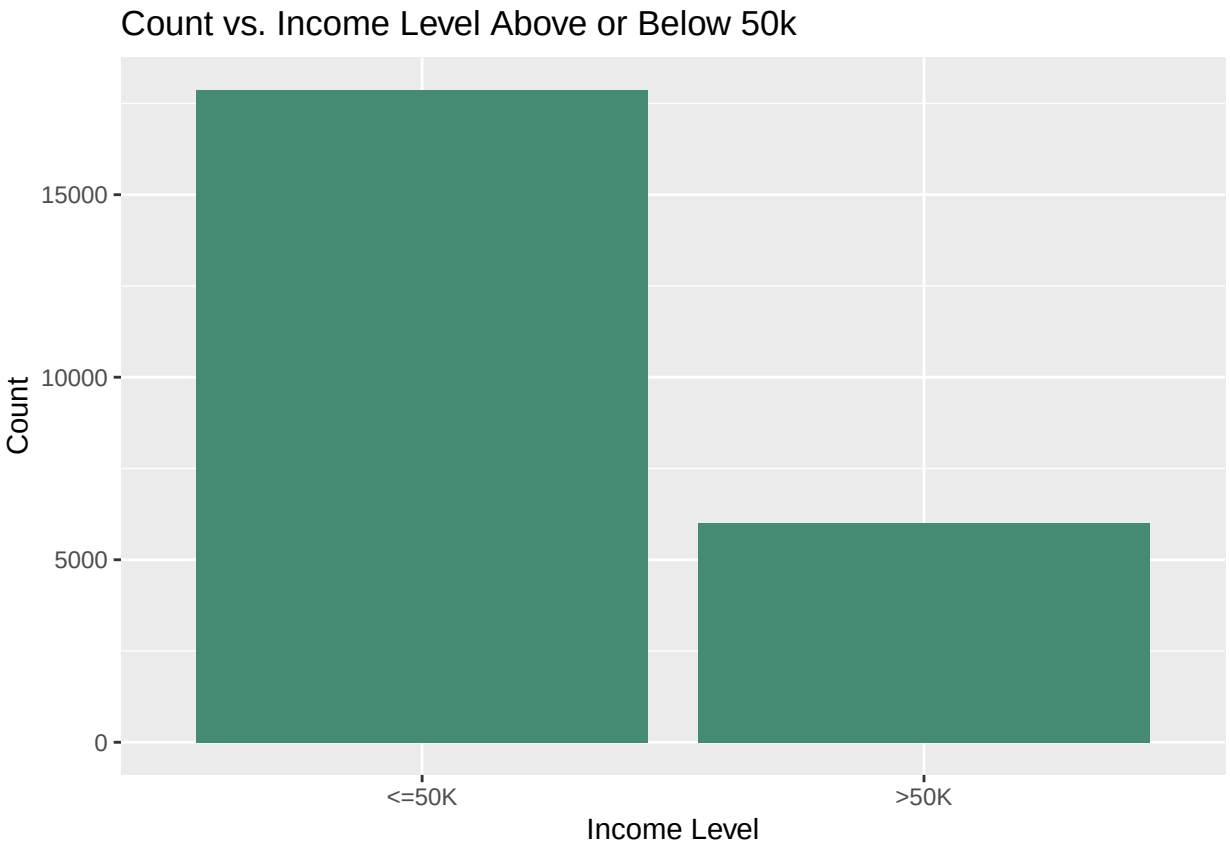

Figure 1: Plot of count versus income level being above or below 50K using train factor data.

Here is the percentage of total individuals in the data set who have an income over 50K.

Table 6: Percentage of Individals with Income Over 50K

x
0.7483449

We can see from both the above graph and table that the data set has more people with income <=50K than >50K. In fact, about 74% of individuals in the data set have income >50K. This is an important element of the data set to note, and I will analyze it in more detail later in this paper.

I next move to analyzing years of education. We can see below a graph of the count vs. education_dummy variable. There are fewer people with very few or very many years of education than there are people in between. For example, the mode of these data is 9 years of education, or roughly a middle school level.

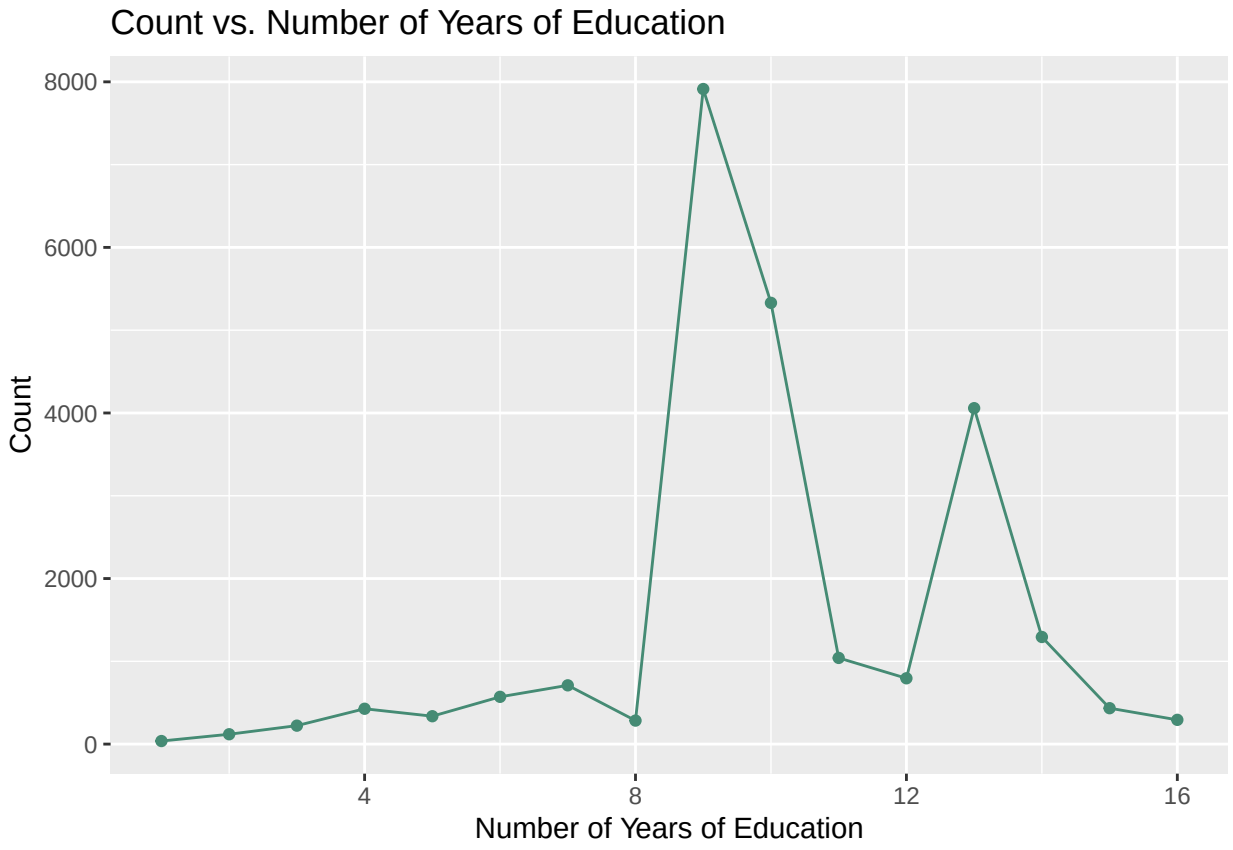

Figure 2: Plot of count versus income level being above or below 50K using train factor data.

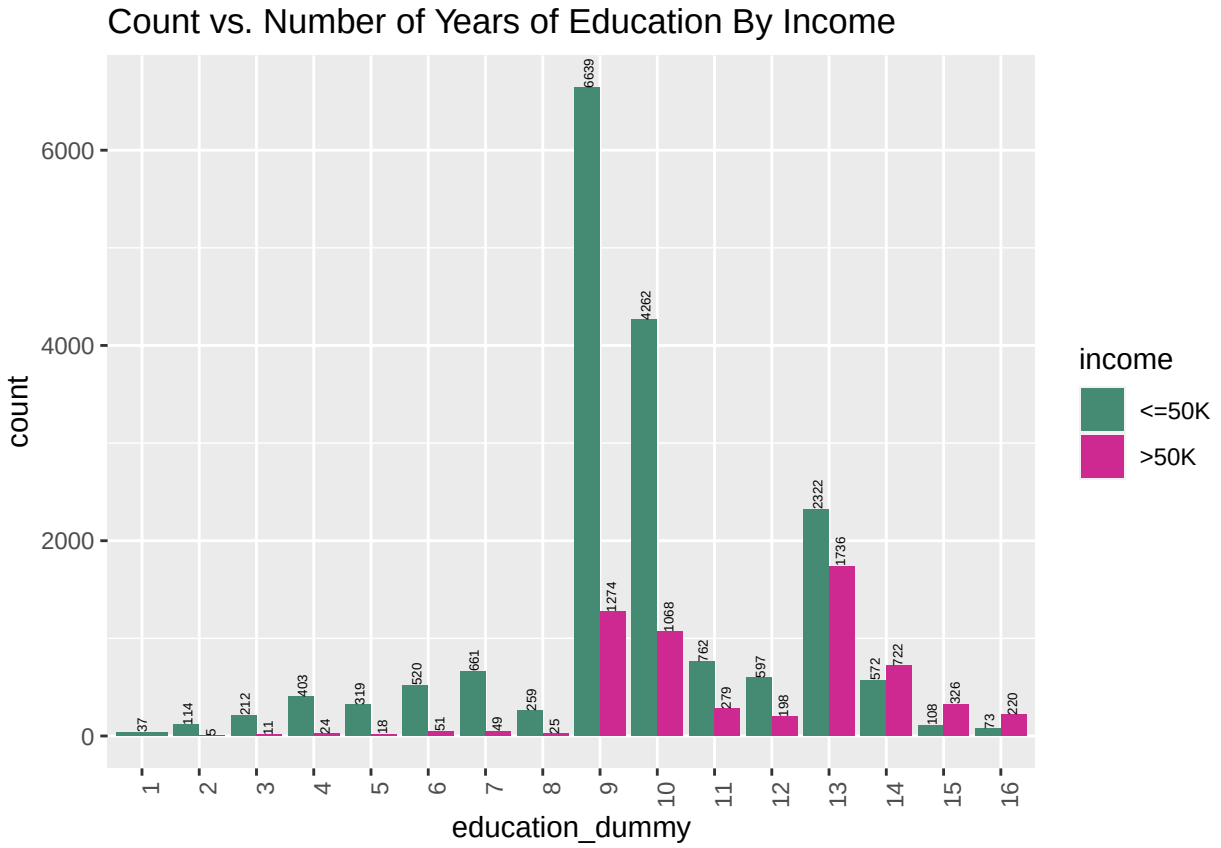

Figure 3: Plot of count versus number of years of education by income level being above or below 50K using train factor data.

Looking at the relationship between years of education and income, people with 14-16 years (a college level) of education have more individuals making >50K than individuals making <=50K. 56% of people with 14 years and 75% of people with 15 or 16 years of education make >50K. 42% of people with 13 years of education (~one year of college) make >50K. Around 20% of people with 9-12 years of education (a high school level) make more than 50K. Only around 5% of people with between 2 and 8 years of education (an elementary and middle school education level) make more than 50K, and people with only 1 year of education have no individuals making more than 50K. This illustrates a logical generalized relationship between education and income – more education leads to a higher income.

Next, I examine the relation between race and income.

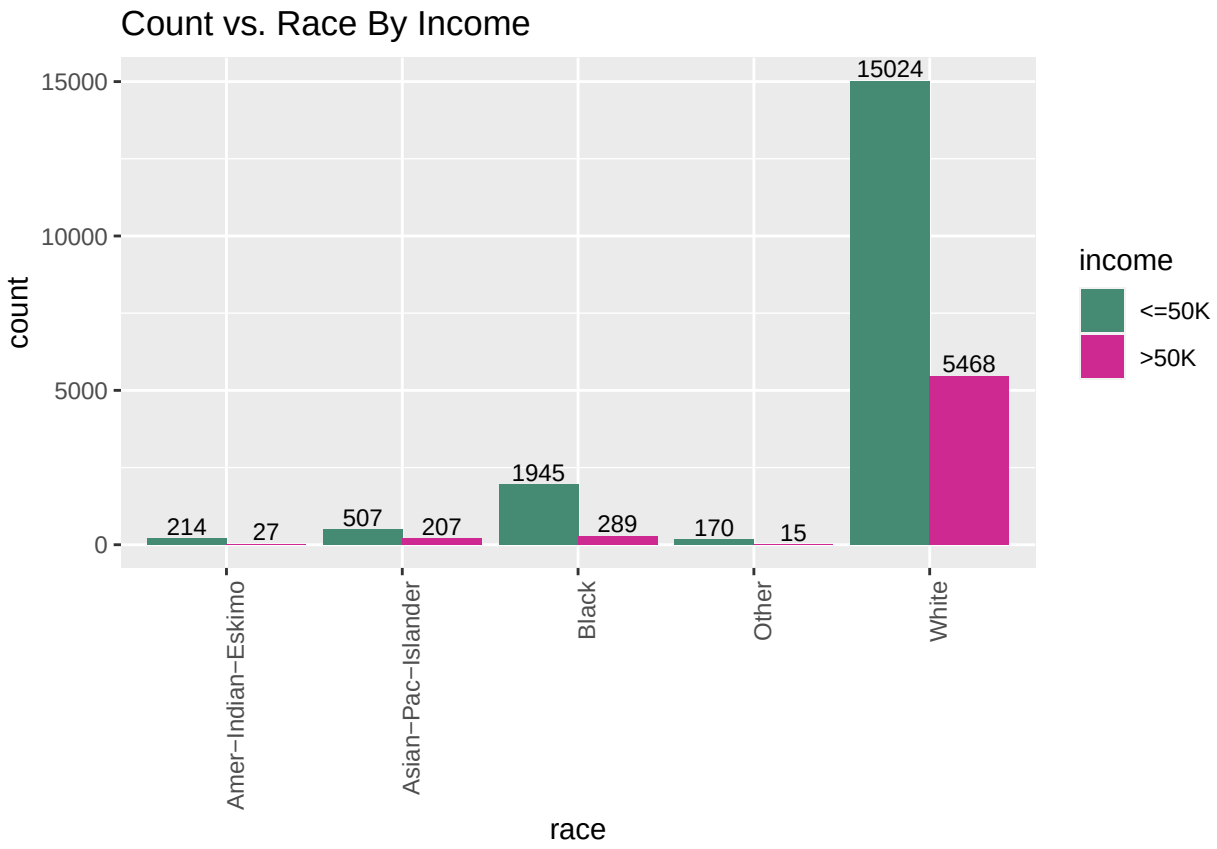

Figure 4: Plot of count versus race by income level being above or below 50K using train factor data.

Here, across race more people are making less than 50K than people making more than 50K. 26% of white people and 28% of Asian/Pacific Islanders made >50K, while only 13% of Black people, 12% of American-Indian/Eskimo people, and 10% of people of other races made >50K. The relationship between race and income in these data, with white people and Asian/Pacific Islanders being more likely to have a higher income in these data than people of other races, holds consistent with other studies showing gaps by race in terms of income level.

I now examine the relation between sex and income.

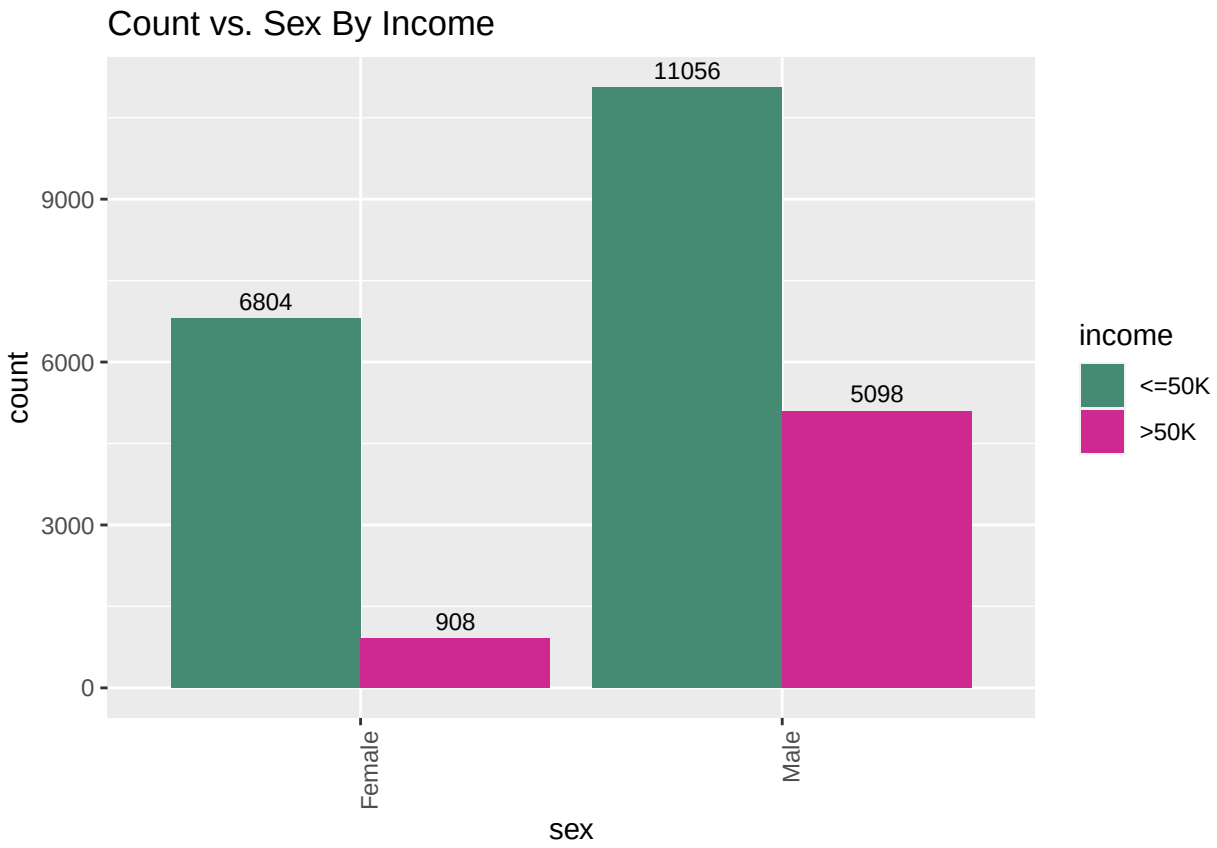

Figure 5: Plot of count versus sex by income level being above or below 50K using train factor data.

Here we can see that, for both men and women, the number of people making less than 50K is higher than those making more than 50K. However, about 1/3 of men make more than 50K, while only around 1/10 of women make more than 50K. This shows that, in these data, men are more likely to have a higher income than women. As with education and race, this also holds consistent with the findings of other analyses about income gaps by education, race, and gender.

Finally, I examine the relationship between marital status and income. For those who are widowed, separated, never married, were married but have an absent spouse, or are divorced, the percent of people making >50K is between about 5 and 10 percent. For people who are married, whether to a civilian or armed forces spouse, about 45% of people have income >50K. In these data, married people are more likely to have a higher income.

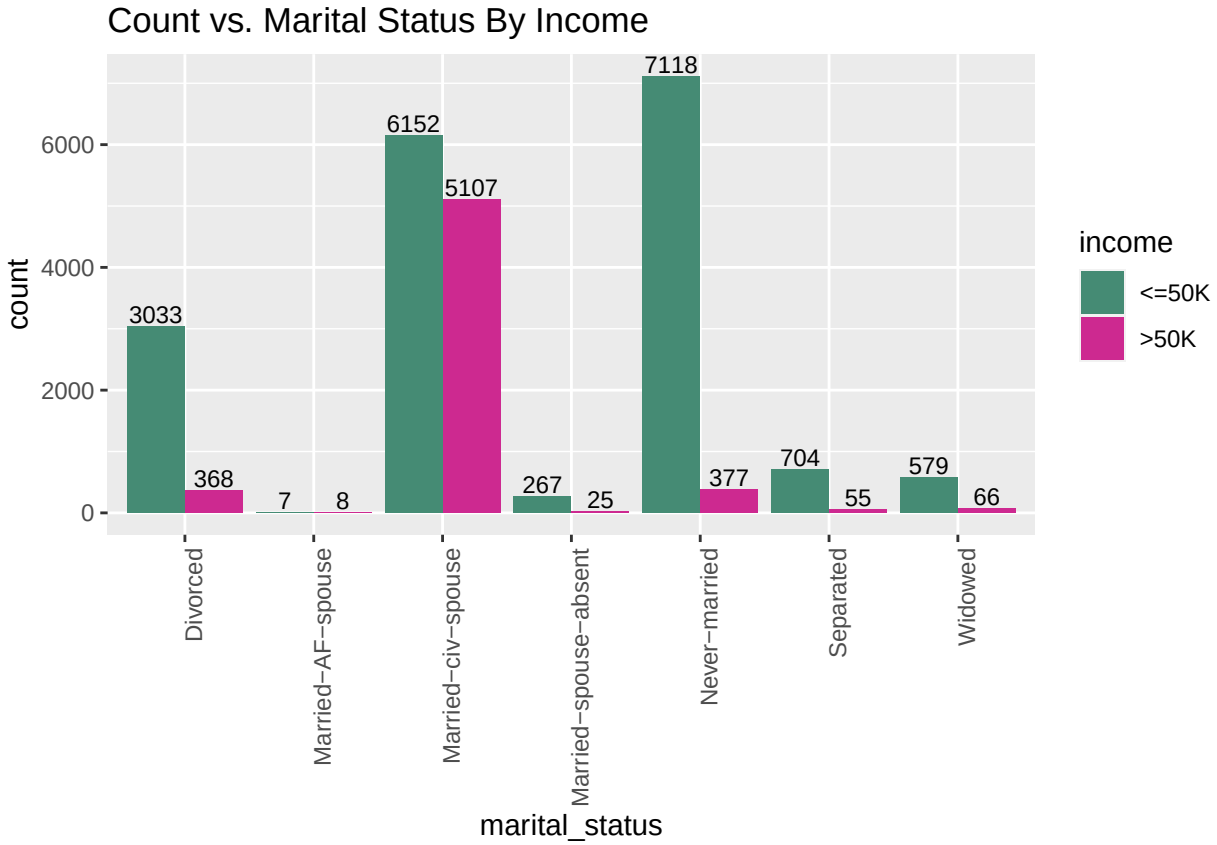

Figure 6: Plot of count versus marital status by income level being above or below 50K using train factor data.

Machine Learning Models

Simplest Model and Understanding Correlation

I now begin developing predictor models. The simplest algorithm is to guess the outcome. This gives an overall accuracy of around 50% as expected. Sensitivity and specificity are also around 50%.

Table 7: Accuracy, Specificity, Sensitivity Results - Including Guessing Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485

Next I create a correlation plot using train_numeric to understand correlation between variables.

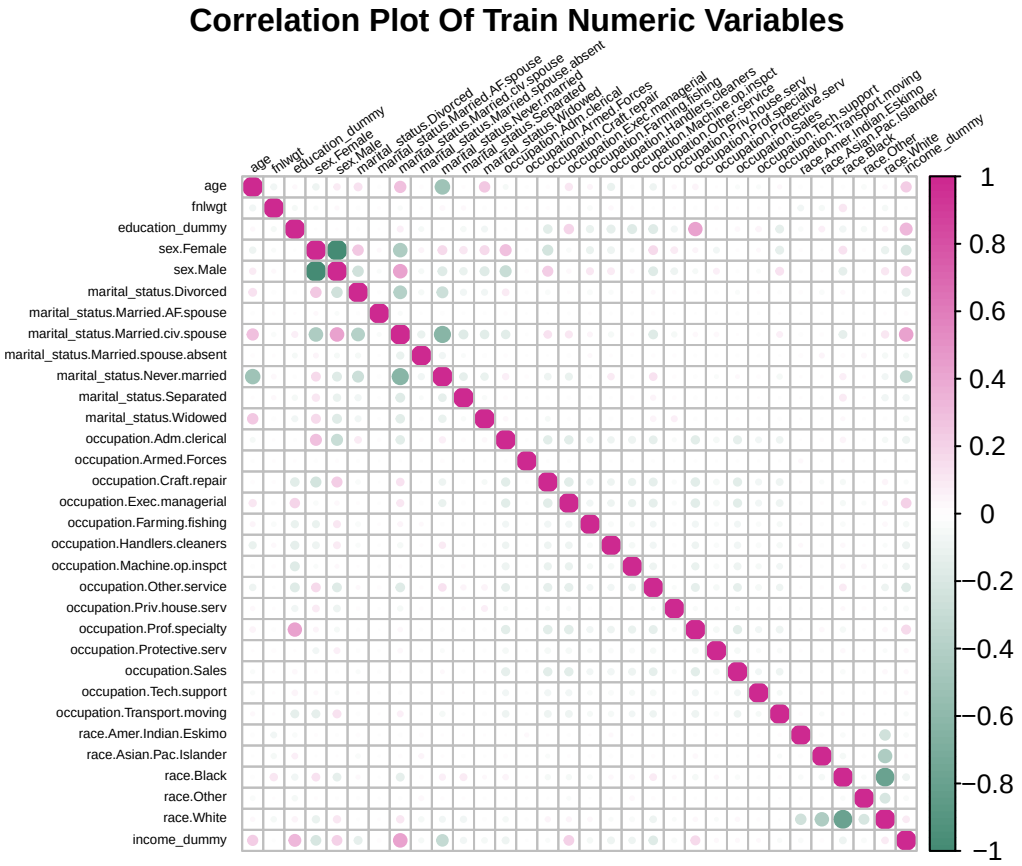

Figure 7: A correlation plot of the train numeric data set.

There appears to be a correlation between several of the variables: marital status and sex; sex and occupation; education and occupation; education and income; marital status and income; and sex and income. Here, we run into a challenge of covariates and multicollinearity. Using correlated coviariates can be challenging. For example, with education by number of years, it logically makes sense that the higher the number of years of education, the higher a person's income level. However, because we have sliced income into a binary variable, once income exceeds 50K we don't glean any more information from education level. With race, because there are 4 covariates (because there are five races recorded, and therefore four covariates), there is more information.

Generalized Linear Models

Now, I move into testing generalized linear models as a machine learning method using the train_factor data In this project, the outcome variable (income) is binary and categorical. The two options are having an income of <=50K or an income >50K. For my next set of models, I apply a logistic regression approach. This can be done using dummy encoding for the outcome (income, Y). In this case, the variable can be encoded through an $ifelse$ call because there is an order to <=50K (smaller, so coded in dummy form as 0) and >50K (larger, so coded in dummy form as 1).

For the predictors, I used a combination of dummy encoding and one-hot encoding. This step was done during the data cleaning process. The choice of whether to use dummy coding or one-hot encoding depends on the predictor. Dummy encoding works only if the categorical variable is ordinal – there is an order to

the outcomes. For instance, education is ordinal because 12 years of education (high school) is more than 8 years of education (middle school). Transitioning the original education variable to years of schooling as the education_dummy variable therefore works, because there is an order to the data. For a category like marital_status, however, there is no order to the categories (other than what one could argue is socially perceived, but that is beyond the scope of this paper). Divorced, separated, widowed, spouse absent, civil marriage, air force marriage, and never married do not have an order, and coding them as such can cause many problems. One-hot encoding is the solution, where each of the categories becomes its own column. For example, there becomes a column for marital_status.Divorced. The values in the column will be 1 if the person was divorced and 0 for all other marital statuses. This is done for each of the categories, forming a full picture of the original marital_status variable.

Logistic regression is a type of generalized linear model (glm). As with a linear regression model, for logistic regression we need the conditional probability of having an income >50K given that a variable X, let's say $X = age$ in this case, is 29 years, $\Pr(Y = 1 \mid X = x)$. In the case of linear and logistic regression, the formula then becomes: $p(x) = \Pr(Y = 1|X = x) = \beta_0 + \beta_1 x$.

However, in logistic regression we contain the values of $\beta_0 + \beta_1 x$ such that they are between 0 and 1, as we are estimating a probability, $p(x)$, which by definition must be between 0 and 1. In logistic regression, therefore, the goal is to find a distribution of Y that has outcomes contained between 0 and 1. Subsequently, the second goal of logistic regression is to find a function g so that $g(\Pr(Y = 1|X = x)$ can be modelled as a linear combination of predictors. This is done using the logistic transformation: $g(p) = \log \frac{p}{1-p}$, which converts the probability to log odds, telling us how much more likely it is for something to happen as opposed to not happening. Therefore, if $p = 0.5$, for instance, the odds are 1:1. Using logistic regression, I model conditional probability using the maximum likelihood estimate (MLE): $g\{\Pr(Y = 1 \mid X = x)\} = \beta_0 + \beta_1 x$. I use the glm function in R to run the logistic regression, but because logistic regression is a type of GLM, I specify logistic regression using the argument family = "binomial". Additionally, to get an output for $\hat{p}(x)$ of the conditional probabilities as opposed to the default logistic transformed values, I specify type = "response". I then use the estimate of $\hat{p}(x)$ to obtain predictions. I set the decision rule here using the boundary as the mean of $\hat{p}$, which in the case of each of the generalized linear models run is approximately 0.25.

Multiple predictors can be used in this model, but I begin by using predictors individually. I start by looking at income and race.

Table 8: Accuracy, Specificity, Sensitivity Results - Including GLM Race Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201

This gives us an overall accuracy of 0.6711, with a sensitivity of 0.8766 and a specificity of 0.0599.

Next, I look at income and sex.

Table 9: Accuracy, Specificity, Sensitivity Results - Including GLM Sex Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201
GLM Sex Model	0.4949732	0.6157635	0.1358189

This gives us an overall accuracy of 0.4950, with a sensitivity of 0.6158 and a specificity of 0.1358.

I then create a generalized linear model using only education. The education model gives us an overall accuracy of 0.2840, with a sensitivity of 0.2203 and a specificity of 0.4734.

Table 10: Accuracy, Specificity, Sensitivity Results - Including GLM Education Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201
GLM Sex Model	0.4949732	0.6157635	0.1358189
GLM Education Model	0.2840147	0.2203314	0.4733688

Next, I create a model looking at income and marital status. The marital status model gives us an overall accuracy of 0.2889, with a sensitivity of 0.3395 and a specificity of 0.1385.

Table 11: Accuracy, Specificity, Sensitivity Results - Including GLM Marital Status Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201
GLM Sex Model	0.4949732	0.6157635	0.1358189
GLM Education Model	0.2840147	0.2203314	0.4733688
GLM Marital Status Model	0.2888740	0.3394536	0.1384820

I now create a model looking at income and occupation. The occupation model gives us an overall accuracy of 0.3396, with a sensitivity of 0.3509 and a specificity of 0.3063.

Table 12: Accuracy, Specificity, Sensitivity Results - Including GLM Occupation Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201
GLM Sex Model	0.4949732	0.6157635	0.1358189
GLM Education Model	0.2840147	0.2203314	0.4733688
GLM Marital Status Model	0.2888740	0.3394536	0.1384820
GLM Occupation Model	0.3396448	0.3508733	0.3062583

Finally, I look at age as a predictor. This gives us an overall accuracy of 0.4204, with a sensitivity of 0.5065 and a specificity of 0.1644.

Table 13: Accuracy, Specificity, Sensitivity Results - Including GLM Age Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201
GLM Sex Model	0.4949732	0.6157635	0.1358189
GLM Education Model	0.2840147	0.2203314	0.4733688
GLM Marital Status Model	0.2888740	0.3394536	0.1384820
GLM Occupation Model	0.3396448	0.3508733	0.3062583
GLM Age Model	0.4204088	0.5064935	0.1644474

Next, to demonstration covariation and multi-collinearity, I create a generalized linear model using all the predictors I used individually above.

Table 14: Accuracy, Specificity, Sensitivity Results - Including GLM All Predictors Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201
GLM Sex Model	0.4949732	0.6157635	0.1358189
GLM Education Model	0.2840147	0.2203314	0.4733688
GLM Marital Status Model	0.2888740	0.3394536	0.1384820
GLM Occupation Model	0.3396448	0.3508733	0.3062583
GLM Age Model	0.4204088	0.5064935	0.1644474
GLM All Predictors Model	0.2144772	0.2342141	0.1557923

This gives us an overall accuracy of 0.2145, with a sensitivity of 0.2342 and a specificity of 0.1558. The results of this logistic model using all predictors gives us the lowest accuracy of all the models we have tried, showing the impact that multi-collinearity has in decreasing model accuracy.

Classification (Decision) Tree Model

The final model I will implement is the classification (decision) tree model. Classification trees (also called decision trees) are used in prediction problems with a categorical outcome, which is the case with income in this project. Classification trees attempt to estimate the conditional expectation $f(x) = \mathrm{E}(Y|X = x)$. The theory of classification trees is to partition the predictor space repeatedly. At a given node, there is a decision rule that determines which side of the tree is followed and, at the end of each node, the goal is to obtain a predictor, $\hat{y}$. In other words, classification trees partition a predictor space J into regions that do not overlap, here represented by $R_1, R_2, \ldots, R_J$. If x, a given predictor, falls in the region R_j then, using the average of the training observations y_i where the associate predictor x_i falls in the region R_j as well, $f(x)$ is estimated. Partitions are recursively created – the entire predictor space is the initial partition that is then split into two partitions, after which one of the two partitions is split again into two, etc.

For classification trees, predictions are formed based on calculations that determine, among each partition in the training set, which class among the observations is most common. Because classification trees use categorical outcomes, choosing the partitions can be done either through a naïve approach, or can be done through more sophisticated methods. The Gini Index is an example of such a method. In a perfect classification tree, in which the Gini Index would equal 0, outcomes in each partition are all of identical categories. The higher the Gini Index, the more we have deviated from the perfect tree. To define the index, first we must define the proportion of observations in a given partition, here termed j, that are of a given class, here termed k. The Gini Index is then: $\mathrm{Gini}(j) = \sum_{k=1}^{K} \hat{p}_{j,k}(1 - \hat{p}_{j,k})$. A second example of a more sophisticated partitioning model is Entropy which is defined, using j and k, as: $\mathrm{entropy}(j) = -\sum_{k=1}^{K} \hat{p}_{j,k} \log(\hat{p}_{j,k})$, with $0 \times \log(0)$ defined as 0.

For the classification (decision) tree I create in this project, I use the train_factor data set and include the dummy and one-hot encoded variables as predictors. I include only $n-1$ of the predictors for a given variable (for instance, the marital_status variable is one-hot encoded into $n = 7$ new variables, of which I include 6). This is due to the fact that, for one-hot encoded variables, the information from $n - 1$ variables is sufficient to determine the output of the n^{th} variable.

I now use the classification (decision) tree model on the data set, including education, sex, age, race, occupation, and marital status as predictor variables. First, I show a plot of the classification (decision) tree.

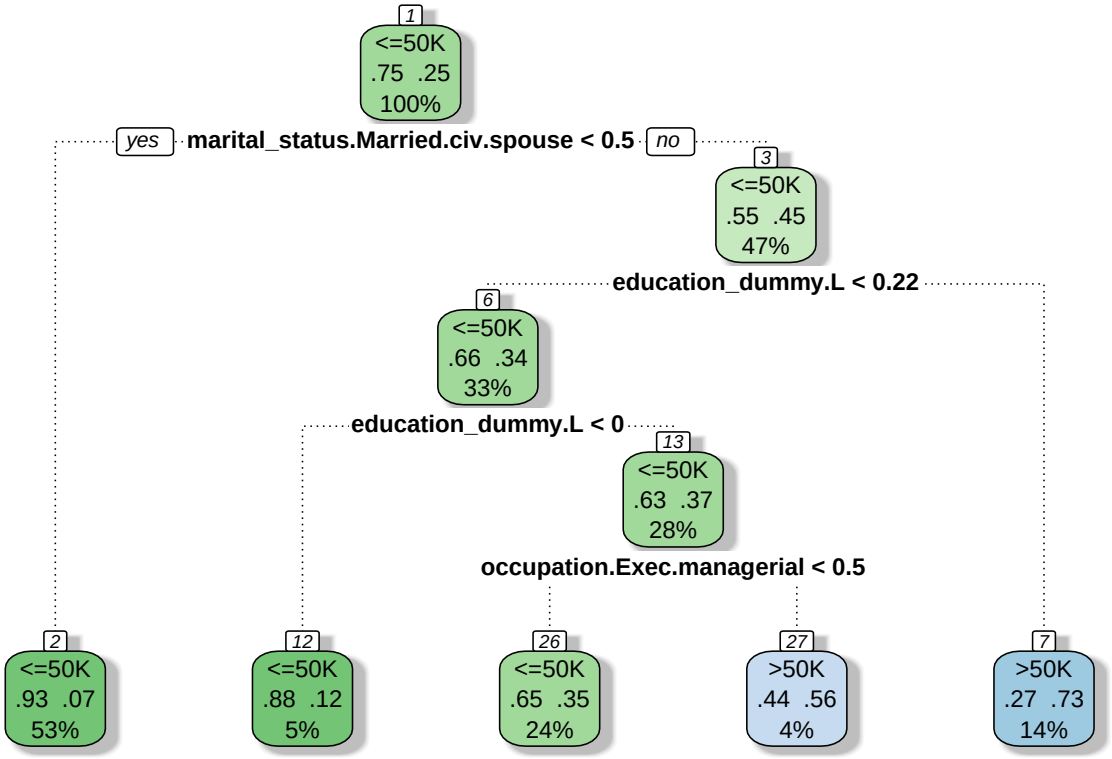

Figure 8: The classification (decision) tree using the train factor data.

I now show a graph of the accuracy vs. complexity parameter for the classification (decision) tree model.

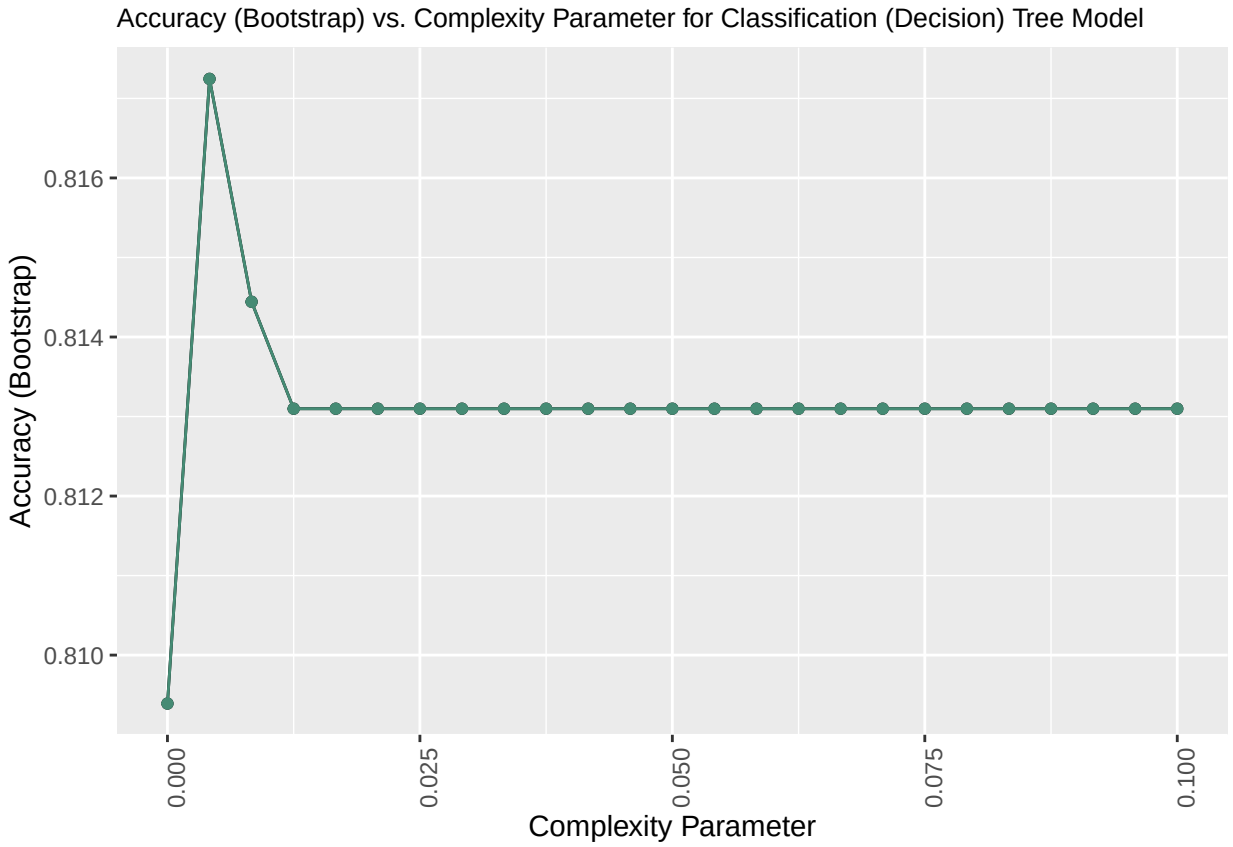

Figure 9: Plot of accuracy (bootstrap) vs. complexity parameter for classification (decision) tree model

In this table, I show the results updated to include the results of the classification (decision) tree model.

Table 15: Accuracy, Specificity, Sensitivity Results - Including Classification (Decision) Tree Model

method	Accuracy	Sensitivity	Specificity
Guessing Model	0.4969832	0.5010638	0.4848485
GLM Race Model	0.6710791	0.8766234	0.0599201
GLM Sex Model	0.4949732	0.6157635	0.1358189
GLM Education Model	0.2840147	0.2203314	0.4733688
GLM Marital Status Model	0.2888740	0.3394536	0.1384820
GLM Occupation Model	0.3396448	0.3508733	0.3062583
GLM Age Model	0.4204088	0.5064935	0.1644474
GLM All Predictors Model	0.2144772	0.2342141	0.1557923
Classification (Decision) Tree Model	0.8183646	0.9308106	0.4840213

The overall accuracy is 0.8184, with a sensitivity of 0.9308 and a specificity of 0.4840.

Results

The results of this analysis show that the most effective prediction model by far is the classification (decision) tree model.

The guessing model performed as expected, with an accuracy of 0.4970, a sensitivity of 0.5010, and a specificity of 0.4848.

Only one of the generalized linear models, the GLM model using race as the sole predictor, performed better than the naive guessing model. However, the specificity of the GLM Race model was only 0.0599. This low specificity implies that the prevalence impacted the model. The GLM sex model (respectively with overall accuracy/sensitivity/specificity of 0.4950/0.6158/0.1358) and GLM age model (respectively, 0.4204/0.5065/0.1644) performed the next best, although they both still had overall accuracy below that of the guessing model and had very low specificity. The occupation model was next, with an overall accuracy of only 0.3396 and sensitivity and specificity of, respectively 0.3509 and 0.3063, followed finally by the marital status model (respectively, 0.2889/0.3395/0.1385) and the education model (respectively, 0.2840/0.2203/0.4734). The fact that marital status and education had the lowest performance is interesting when considering that they were two of the variables more correlated with income. The GLM model with all predictors performed the worst (respectively, 0.2145/0.2342/0.1558), which makes sense given the covariation and multi-collinearity of the various predictor variables.

The classification (decision) tree model allowed for the use of multiple predictors in decision tree format without running into the same challenges as with the generalized linear models (logistic regression). The classification tree model had an accuracy of 0.8184, sensitivity of 0.9308, and specificity of 0.4840. This model shows to be especially effective given that the accuracy and sensitivity are both relatively high, and the specificity is moderately high as well.

Conclusion

For my final project for the Harvard EdX professional certificate in data science, I set out to develop a prediction model to determine if an individuals income was >50k or <=50k. I began by pre-processing and cleaning the data – removing unnecessary columns and rows that were missing data. I dummy encoded and one-hot encoded the variables for use in my models and then I proceeded to develop a numeric data set and a factor data set that included both numeric and factor variables. Using the same index, I then partitioned both the numeric and factor data sets respectively into test sets (20%) and train sets (80%). The partitions were created such that I ended up with the data sets train_numeric, test_numeric, train_factor, and test_factor. I then visualized the data, before beginning to test a series of models. I first tested the naïve guessing model, before testing a series of generalized linear models (logistic) using various predictors. I finished by running a classification (decision) tree model on the data. For the generalized linear models (logistic), I found that, as expected, using more predictors *lowered* the overall accuracy. I predict that this was due to covariation and multi-collinearity. Of the generalized linear models, the GLM Race model performed the best, with an overall accuracy of 0.6681, a sensitivity of 0.8744, and a specificity of 0.0546. The model in which I used all predictors performed the worst, with an overall accuracy of 0.2145, with a sensitivity of 0.2342 and a specificity of 0.1558. The most successful model I tested was the classification (decision) tree model, which had an overall accuracy was 0.8184, sensitivity of 0.9308, and specificity of 0.4840. This model has relatively high overall accuracy and sensitivity, and moderately high specificity, making it by far the most effective model tested.

This report verifies findings that activists and social scientists have long known about and have been working to combat, namely the relation of race, sex, marital status, age, etc. to income level. These findings are important in continuing to recognize these problematic phenomena and work to change them using data-based evidence. This predictor model specifically could also be useful to such activist, non-profit, or philanthropic groups, for instance to recognize the probability of a given individuals monetary status when deciding where to allocate vital and limited resources.

While this model performed well, there is much future work that could be done to expand and improve this report. Although one-hot encoding is effective, it can also lead to increased multi-collinearity. Using a different method of encoding variables could therefore improve results, but was outside the scope of this project. In addition, models could be run using some of the variables that I removed due to the time and processing constraints of this project. Additional models could also be run to gain greater insight into the data and to increase performance. Ron Kohavi, the developer of this data set, runs an NBTrees model, for instance, which combines naïve-bayes with the decision tree models. I would be interested to test such models in future expansions of this project.

Sources

- Dalpiaz, D. (2020, October 28). R for Statistical Learning. https://daviddalpiaz.github.io/r4sl/.
- Data.World (2017). Census Income. https://data.world/uci/census-income/workspace/file?filename=adult.data.csv.
- Grace-Martin, K. (2019). Interpreting the Intercept in a Regression Model. https://www.theanalysisfactor.com/interpreting-the-intercept-in-a-regression-model/.
- Irizarry, R. A. (2020, March 2). Introduction to Data Science. Retrieved from https://rafalab.github.io/dsbook/introduction-to-machine-learning.html#notation-1.
- Kaggle.com (2016). Adult Census Income. https://www.kaggle.com/uciml/adult-census-income?select=adult.csv.
- Kohavi, R. (1996). "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. http://robotics.stanford.edu/~ronnyk/nbtree.pdf.
- LaMorte, W. W. (2018, February 26). Dummy Variables in R. https://sphweb.bumc.bu.edu/otlt/MPH-Modules/PH717-QuantCore/PH717_MultipleVariableRegression/PH717_MultipleVariableRegression4.html.
- Lichman, M. (2013). UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets/Census+Income. Irvine, CA: University of California, School of Information and Computer Science.
- Mahto, K. K. (2019, July 8). One-Hot-Encoding, Multicollinearity and the Dummy Variable Trap. https://towardsdatascience.com/one-hot-encoding-multicollinearity-and-the-dummy-variable-trap-b5840be3c41a.
- Oehm, D. (2018, June 7). Design Matrix for Regression Explained. http://gradientdescending.com/design-matrix-for-regression-explained/.
- United States Census Bureau, (1996, April 1). Income, Poverty, and Valuation of Noncash Benefits: 1994. https://www.census.gov/library/publications/1996/demo/p60-189.html.