HarvardX PH125.9x Data Science Capstone Adults Census Income Project

Amber Cavasos

2024-04-25

Introduction

This project aims to leverage the data science skills acquired through the HarvardX Data Science program to explore, cleanse, and analyze data to uncover patterns and insights, and to ultimately construct a predictive model. The project serves as a practical application of data science principles, offering a hands-on approach to understanding and manipulating data to derive meaningful conclusions and predictions. It represents an opportunity to reinforce learning by applying theoretical knowledge in a real-world context.

Dataset

For my final capstone project, I chose the “Adult Census Income” dataset. The dataset, sourced from the 1994 U.S. Census Bureau database, can help in the development of a prediction model to determine whether an individual earns more than $50,000 per year based on demographic and employment data. This dataset includes attributes such as age, workclass, education level, marital status, occupation, race, gender, native country, hours worked per week, and more.

Methods and Analysis

In my final capstone project, I conducted a comprehensive analysis of the UCI Adult dataset, beginning with necessary package management and data loading. I progressed through data wrangling— cleansing and formatting the data, and perform exploratory data analysis, using visualizations to examine relationships between demographic and socio-economic variables and income. The data is then partitioned into training and testing sets. Various predictive models including logistic regression, random forest, and a classification tree are constructed and evaluated on their accuracy. I also detail the model evaluation on the final holdout set to assess their generalization capability, concluding with a compilation of model performances, highlighting the effectiveness of each modeling approach in predicting income levels.

Loading Data

```
data_url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"

data <- read_csv(data_url, col_names = c("age", "working_class", "final_weight", "education", "educatio
                                          "occupation", "relationship", "race", "gender", "capital_gai
                                          "native_country", "income"))
```

```
## Rows: 32561 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (9): working_class, education, marital_status, occupation, relationship,...
## dbl (6): age, final_weight, education_num, capital_gain, capital_loss, hours...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

Data Wrangling

This process involves converting and restructuring data from its original raw state into a more useful format. The primary objective of data wrangling is to ensure the data is of high quality and utility for its intended uses.

Lets look at the first few rows of data.

```
head(data)
```

```
## # A tibble: 6 x 15
##     age working_class    final_weight education education_num marital_status
##   <dbl> <chr>                   <dbl> <chr>             <dbl> <chr>
## 1    39 State-gov               77516 Bachelors            13 Never-married
## 2    50 Self-emp-not-inc        83311 Bachelors            13 Married-civ-spouse
## 3    38 Private                215646 HS-grad               9 Divorced
## 4    53 Private                234721 11th                  7 Married-civ-spouse
## 5    28 Private                338409 Bachelors            13 Married-civ-spouse
## 6    37 Private                284582 Masters              14 Married-civ-spouse
## # i 9 more variables: occupation <chr>, relationship <chr>, race <chr>,
## #   gender <chr>, capital_gain <dbl>, capital_loss <dbl>, hours_per_week <dbl>,
## #   native_country <chr>, income <chr>
```

Data Exploration

The dataset is structured as a tibble containing 32,561 rows and 15 columns. Each row corresponds to a specific group of individuals with similar preferences, while each column provides various pieces of personal information about these people.

```
dim(data)
```

```
## [1] 32561    15
```

```
str(data)
```

```
## spc_tbl_ [32,561 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ age           : num [1:32561] 39 50 38 53 28 37 49 52 31 42 ...
##  $ working_class : chr [1:32561] "State-gov" "Self-emp-not-inc" "Private" "Private" ...
##  $ final_weight  : num [1:32561] 77516 83311 215646 234721 338409 ...
##  $ education     : chr [1:32561] "Bachelors" "Bachelors" "HS-grad" "11th" ...
##  $ education_num : num [1:32561] 13 13 9 7 13 14 5 9 14 13 ...
##  $ marital_status: chr [1:32561] "Never-married" "Married-civ-spouse" "Divorced" "Married-civ-spouse
##  $ occupation    : chr [1:32561] "Adm-clerical" "Exec-managerial" "Handlers-cleaners" "Handlers-clea
##  $ relationship  : chr [1:32561] "Not-in-family" "Husband" "Not-in-family" "Husband" ...
##  $ race          : chr [1:32561] "White" "White" "White" "Black" ...
##  $ gender        : chr [1:32561] "Male" "Male" "Male" "Male" ...
##  $ capital_gain  : num [1:32561] 2174 0 0 0 0 ...
##  $ capital_loss  : num [1:32561] 0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: num [1:32561] 40 13 40 40 40 40 16 45 50 40 ...
##  $ native_country: chr [1:32561] "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr [1:32561] "<=50K" "<=50K" "<=50K" "<=50K" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   working_class = col_character(),
##   ..   final_weight = col_double(),
##   ..   education = col_character(),
##   ..   education_num = col_double(),
##   ..   marital_status = col_character(),
##   ..   occupation = col_character(),
##   ..   relationship = col_character(),
##   ..   race = col_character(),
##   ..   gender = col_character(),
##   ..   capital_gain = col_double(),
##   ..   capital_loss = col_double(),
##   ..   hours_per_week = col_double(),
##   ..   native_country = col_character(),
##   ..   income = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Data Cleaning

The primary objective of data cleaning is to eliminate errors and inconsistencies that can compromise data quality and analysis outcomes. This careful preparation helps maximize the data's utility for its intended analytical or operational purposes.

There appear to be missing values.

```
anyNA(data)
```

```
## [1] FALSE
```

Here, we can see that some rows have "?" values in following features.

```
unique(data$working_class)
```

```
## [1] "State-gov"        "Self-emp-not-inc" "Private"          "Federal-gov"
## [5] "Local-gov"        "?"                "Self-emp-inc"     "Without-pay"
## [9] "Never-worked"
```

```
unique(data$occupation)
```

```
##  [1] "Adm-clerical"      "Exec-managerial"   "Handlers-cleaners"
##  [4] "Prof-specialty"    "Other-service"     "Sales"
##  [7] "Craft-repair"      "Transport-moving"  "Farming-fishing"
## [10] "Machine-op-inspct" "Tech-support"      "?"
## [13] "Protective-serv"   "Armed-Forces"      "Priv-house-serv"
```

```
unique(data$native_country)
```

```
##  [1] "United-States"                "Cuba"
##  [3] "Jamaica"                      "India"
##  [5] "?"                            "Mexico"
##  [7] "South"                        "Puerto-Rico"
##  [9] "Honduras"                     "England"
## [11] "Canada"                       "Germany"
## [13] "Iran"                         "Philippines"
## [15] "Italy"                        "Poland"
## [17] "Columbia"                     "Cambodia"
## [19] "Thailand"                     "Ecuador"
## [21] "Laos"                         "Taiwan"
## [23] "Haiti"                        "Portugal"
## [25] "Dominican-Republic"           "El-Salvador"
## [27] "France"                       "Guatemala"
## [29] "China"                        "Japan"
## [31] "Yugoslavia"                   "Peru"
## [33] "Outlying-US(Guam-USVI-etc)" "Scotland"
## [35] "Trinadad&Tobago"              "Greece"
## [37] "Nicaragua"                    "Vietnam"
## [39] "Hong"                         "Ireland"
## [41] "Hungary"                      "Holand-Netherlands"
```

Let’s change them to “Other”.

```
data <- data %>%
  mutate(
    working_class = if_else(working_class == "?", "Other", working_class),
    occupation = if_else(occupation == "?", "Other", occupation),
    native_country = if_else(native_country == "?", "Other", native_country)
  )
```

Here are the dimensions after this change.

```
dim(data)
```

```
## [1] 32561    15
```

Remove Unecessary Variables

We can remove the “fnlwgt” variable stands for “final weight.” This value represents the number of people the census believes the entry corresponds to, based on the demographic characteristics of the person. Essentially, this weight is calculated to ensure that the dataset is a representative sample of the U.S. population, but is not needed for this project. The “education” variable is redundant, as we also have “educatio.num”.

```
data <- data %>% select(-final_weight, -education)
```

After cleaning the data set, we are left with 13 columns.

```
str(data)
```

```
## tibble [32,561 x 13] (S3: tbl_df/tbl/data.frame)
##  $ age           : num [1:32561] 39 50 38 53 28 37 49 52 31 42 ...
##  $ working_class : chr [1:32561] "State-gov" "Self-emp-not-inc" "Private" "Private" ...
##  $ education_num : num [1:32561] 13 13 9 7 13 14 5 9 14 13 ...
##  $ marital_status: chr [1:32561] "Never-married" "Married-civ-spouse" "Divorced" "Married-civ-spouse
##  $ occupation    : chr [1:32561] "Adm-clerical" "Exec-managerial" "Handlers-cleaners" "Handlers-clea
##  $ relationship  : chr [1:32561] "Not-in-family" "Husband" "Not-in-family" "Husband" ...
##  $ race          : chr [1:32561] "White" "White" "White" "Black" ...
##  $ gender        : chr [1:32561] "Male" "Male" "Male" "Male" ...
##  $ capital_gain  : num [1:32561] 2174 0 0 0 0 ...
##  $ capital_loss  : num [1:32561] 0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: num [1:32561] 40 13 40 40 40 40 16 45 50 40 ...
##  $ native_country: chr [1:32561] "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr [1:32561] "<=50K" "<=50K" "<=50K" "<=50K" ...
```

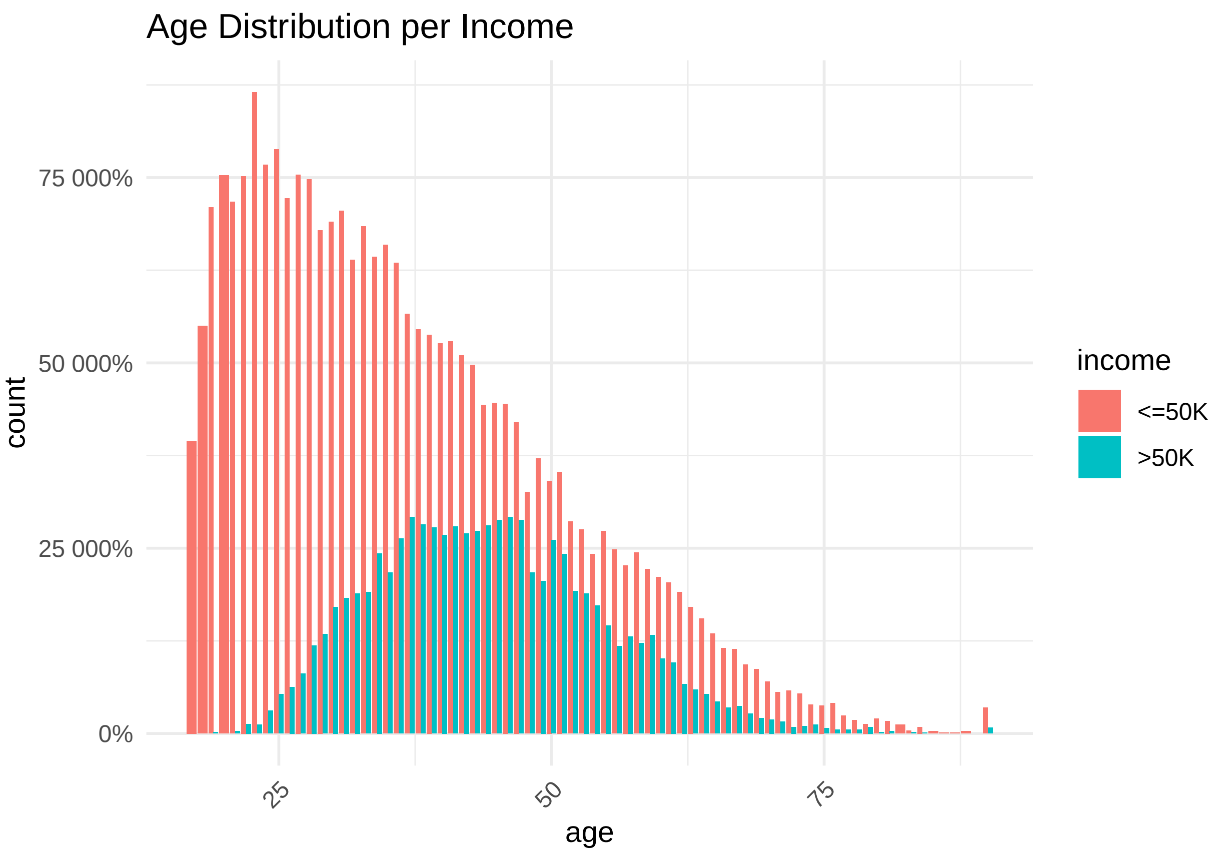
Exploratory Data Analysis

Exploratory Data Analysis is the process of examining datasets to summarize their key features, often through visual methods. This step is crucial for gaining insights into the data and understanding its underlying patterns and relationships.

Age vs Income

Let's begin by identifying patterns and trends that might indicate the ages at which individuals are most likely to earn more than $50,000 annually.

```
data %>%
  ggplot(aes(x = age, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Age Distribution per Income") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

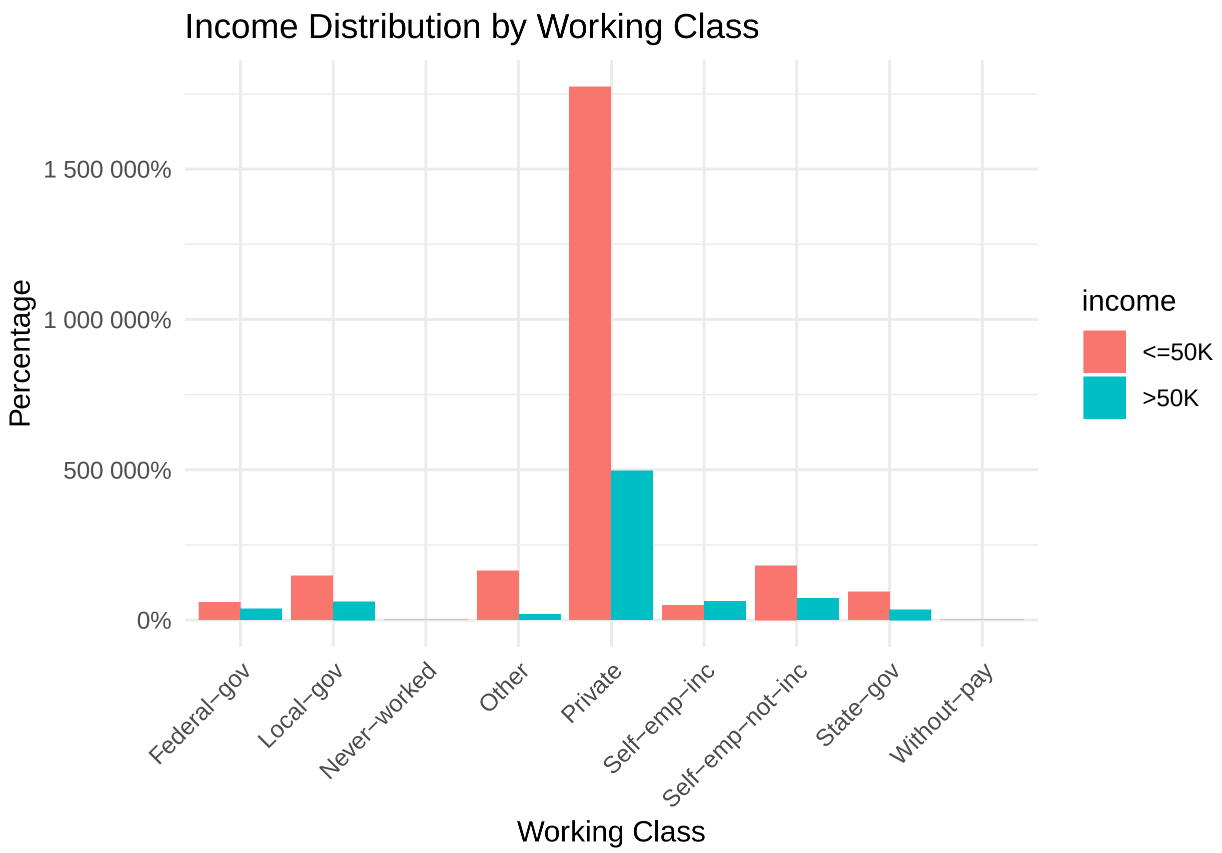


Most people who earn more than 50k are between 30 and 60 years.

Workclass vs Income

Next, let's look at how different employment sectors correlate with the likelihood of earning more than $50,000 per year.

```r
data %>%
  ggplot(aes(x = working_class, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Income Distribution by Working Class", x = "Working Class", y = "Percentage") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

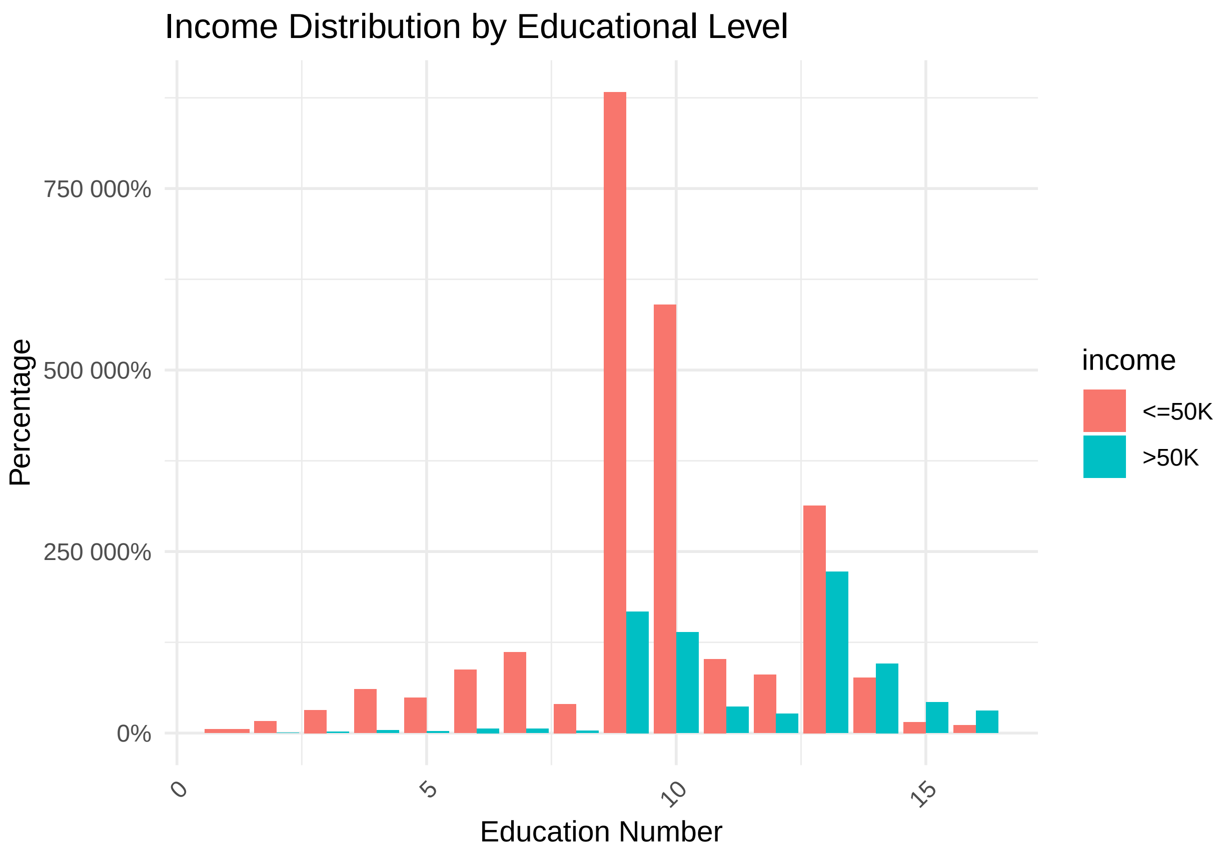


Most of the data is in the “Private” group.

Education.num vs Income

The variable “Education Number” represents the level of education, ranging from 1 (Preschool) to 16 (Doctorate).

```
data %>%
  ggplot(aes(x = education_num, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Income Distribution by Educational Level", x = "Education Number", y = "Percentage") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

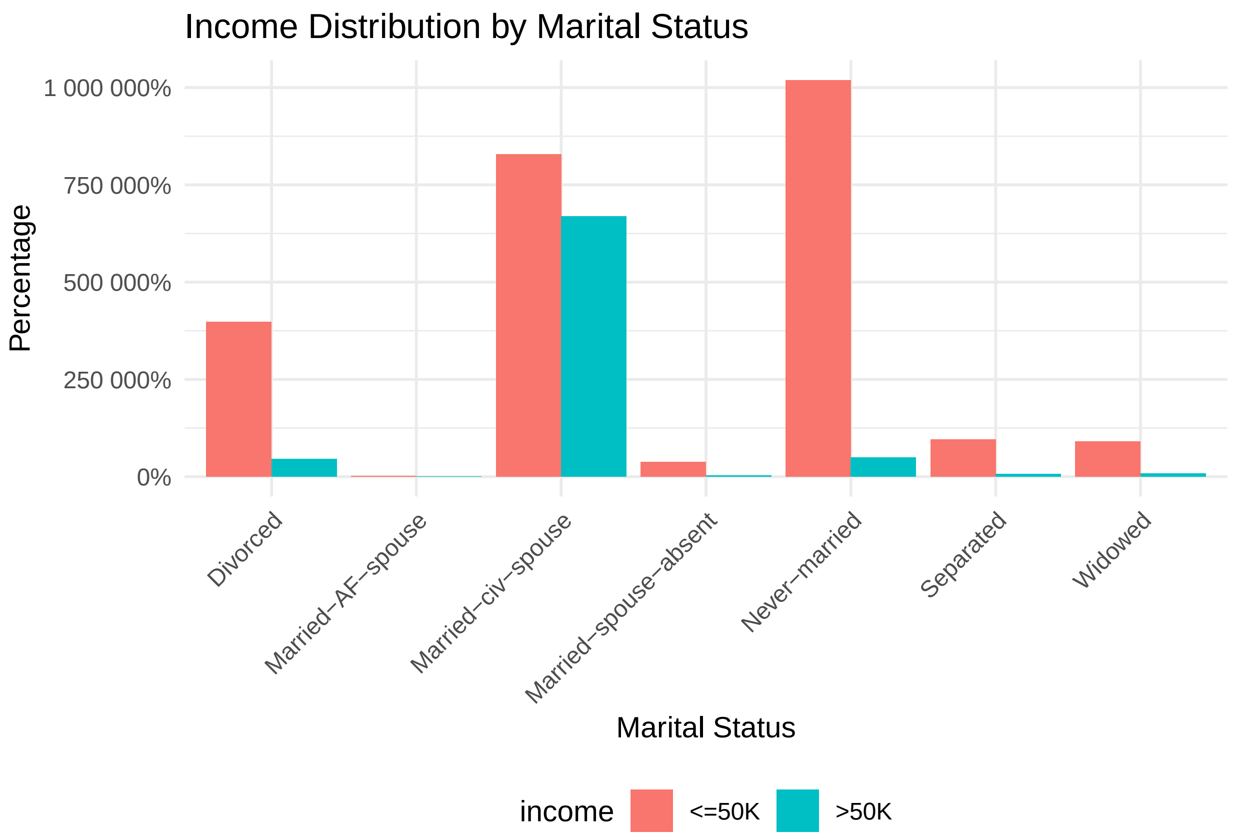


As the level of education increases, so does the percentage of individuals earning an income above 50k.

Marital Status vs Income

How does marital status correlate with income levels?

```r
data %>%
  ggplot(aes(x = marital_status, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Income Distribution by Marital Status", x = "Marital Status", y = "Percentage") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "bottom"
  )
```

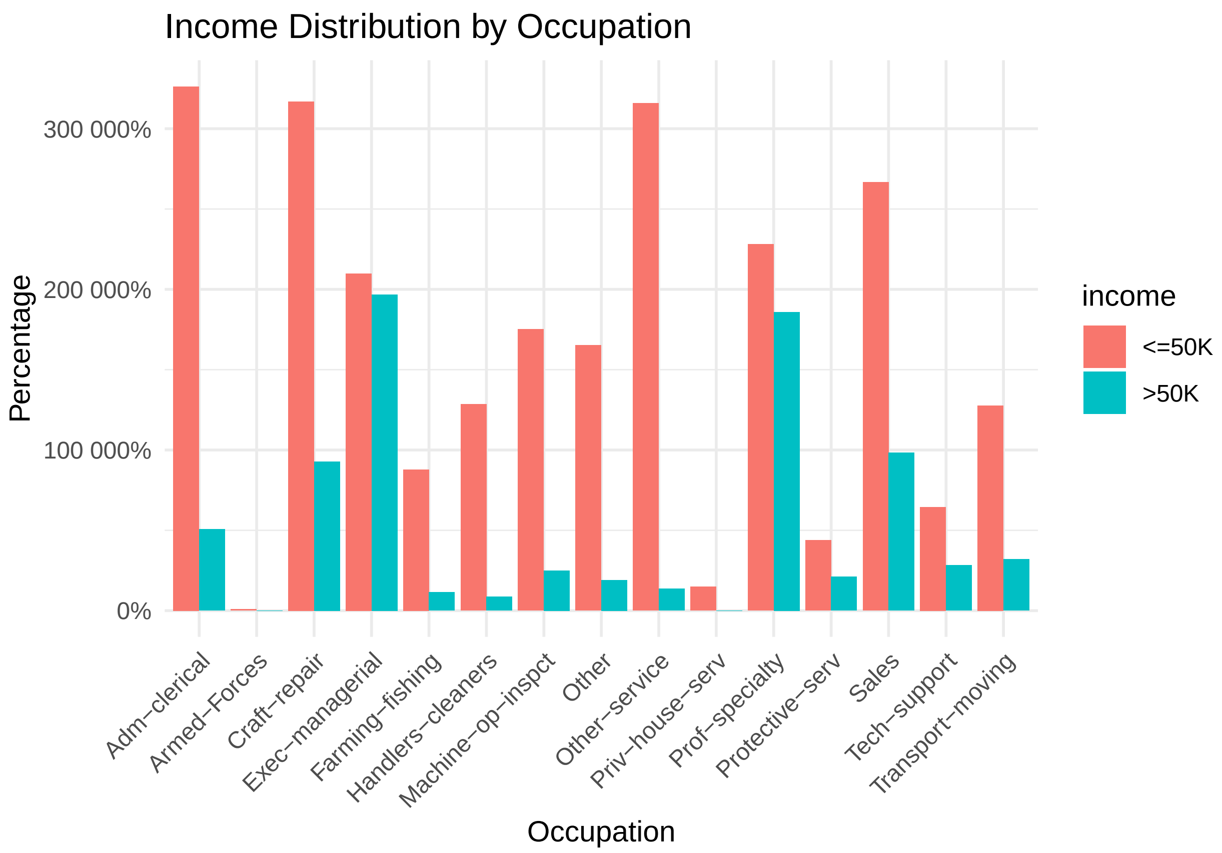


The distribution of individuals earning over 50k in income across different marital statuses is relatively even, with the exception of those identified as “Married-civ-spouse” (which refers to an individual who is married to a spouse who is a civilian) and those identified as “Married-AF-spouse” (which refers to an individual who is married to a spouse who is in the armed forces). Generally, those who are married tend to make much more than those who are not.

Occupation vs Income

The 14 occupations featured in this dataset are: 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, and the Armed-Forces.

```r
data %>%
  ggplot(aes(x = occupation, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Income Distribution by Occupation", x = "Occupation", y = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

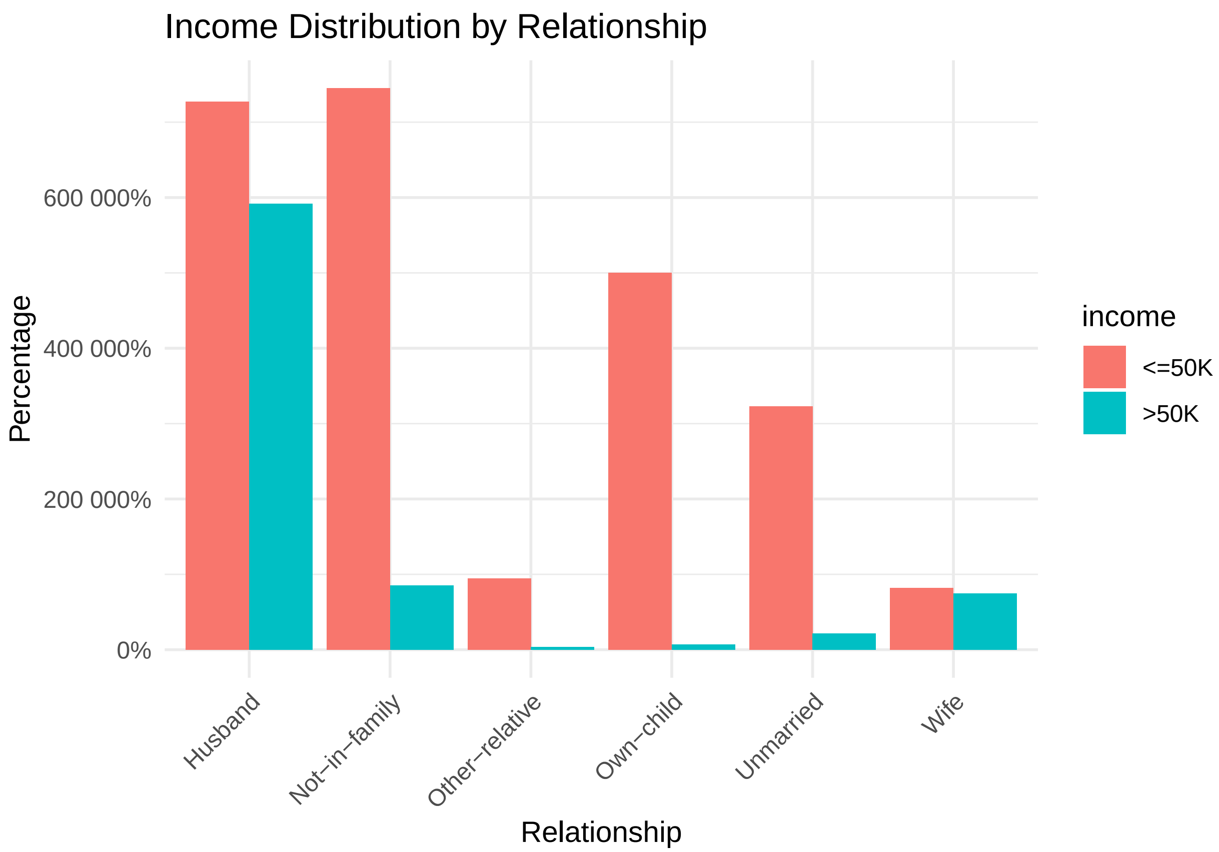


Some occupations have a higher percentage of individuals earning over 50k, especially Exec-managerial, and Prof-speciailty.

Relationship vs Income

This indicates the person's relationship status, which can be categorized into six distinct categories: husband, not-in-family, other-relative, own-child, unmarried, wife.

```r
data %>%
  ggplot(aes(x = relationship, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Income Distribution by Relationship", x = "Relationship", y = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

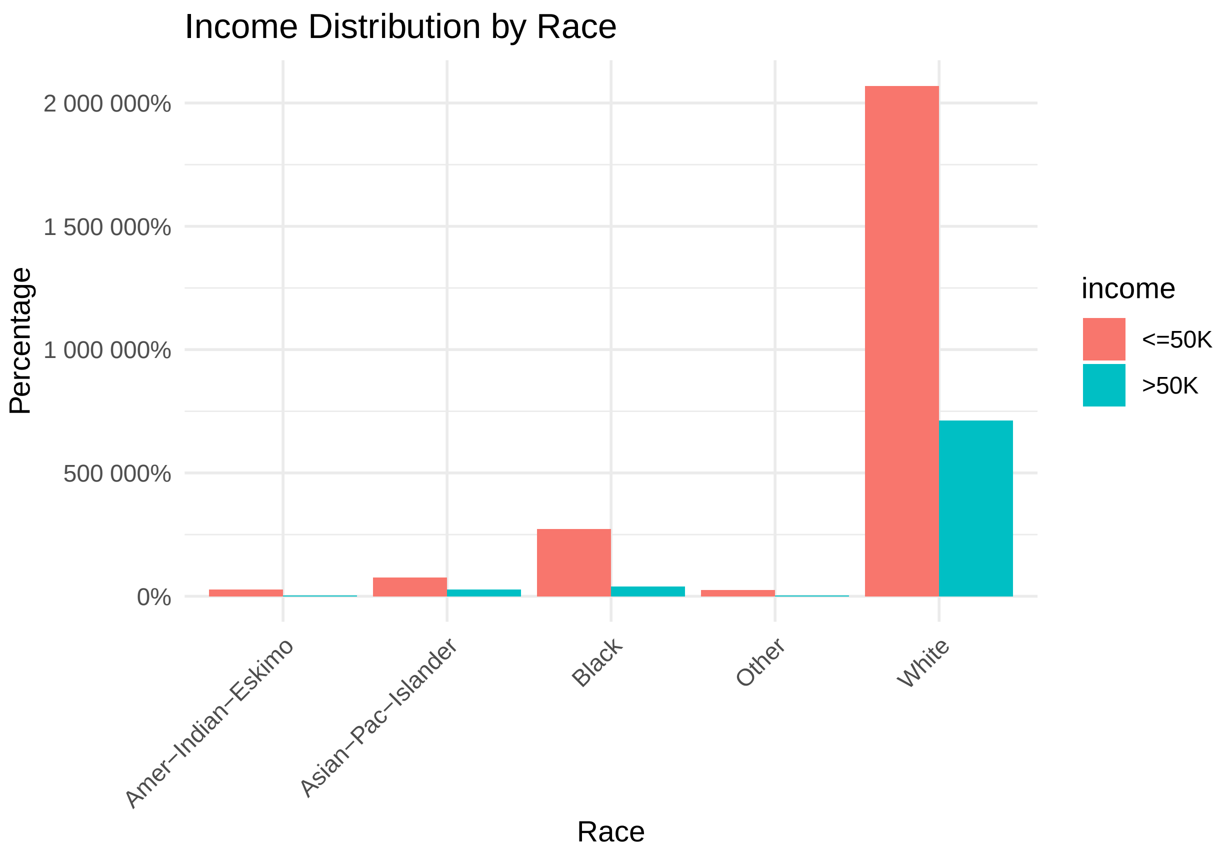


This agrees with the observations noted in the marital status analysis: Married people earn more than those who are not married.

Race vs Income

Let's examine the relationship between an individual's race and their income levels.

```
data %>%
  ggplot(aes(x = race, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Income Distribution by Race", x = "Race", y = "Percentage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

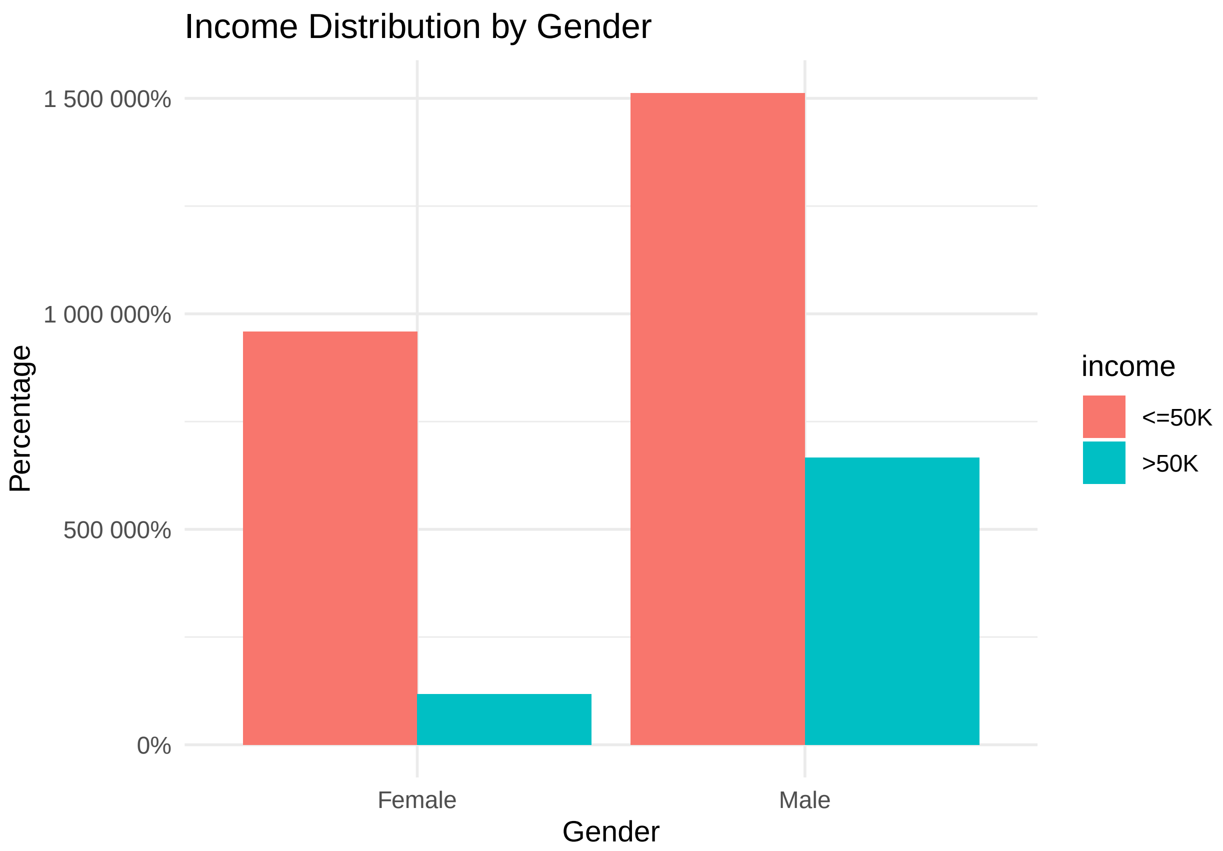


Nearly all individuals with an income exceeding $50,000 are white.

Gender vs Income

Let's explore how income levels are distributed across different genders.

```r
data %>%
  ggplot(aes(x = gender, fill = income)) +
  geom_bar(position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Income Distribution by Gender", x = "Gender", y = "Percentage") +
  theme_minimal()
```

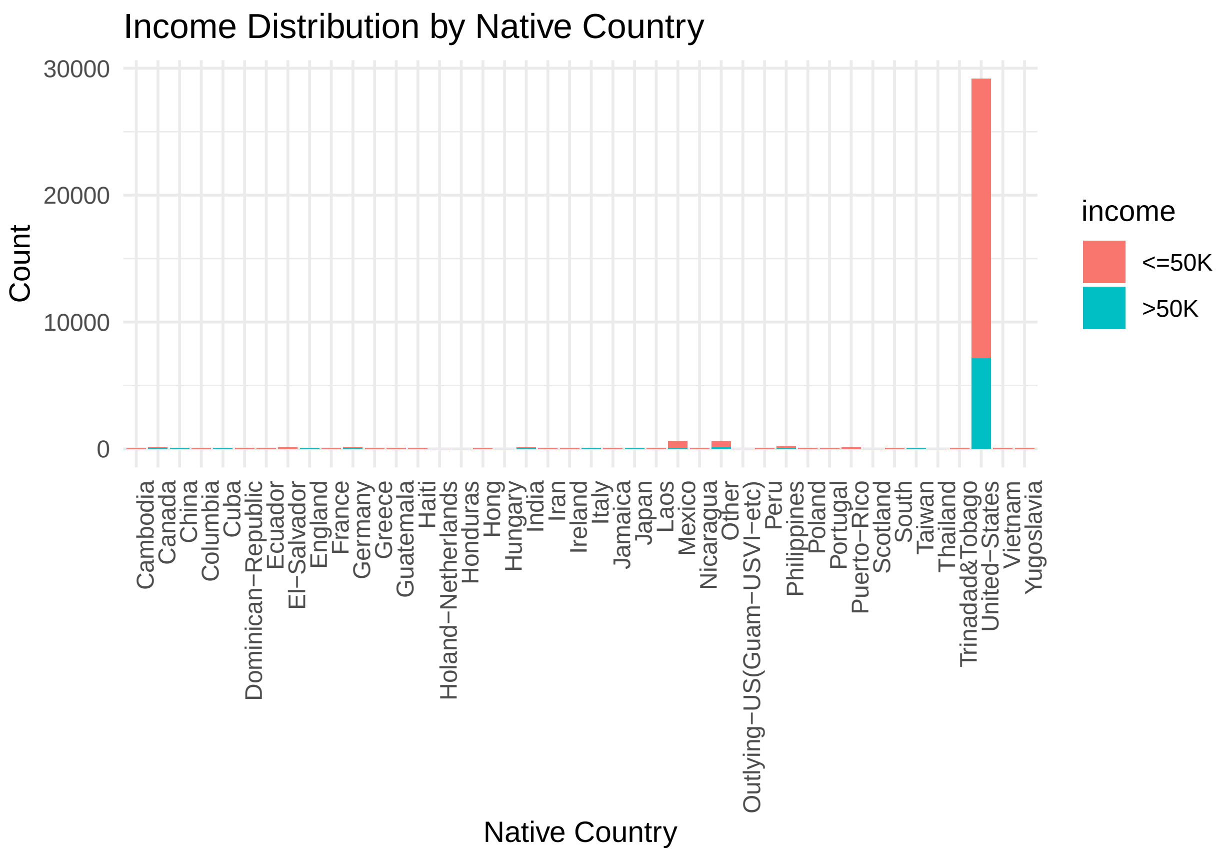


The majority of individuals earning over $50,000 are male.

Native Country vs Income

Finally, let's look at the country of origins for participants.

```
data %>%
  ggplot(aes(x = native_country, fill = income)) +
  geom_bar(position = "stack") +
  scale_y_continuous() +
  labs(title = "Income Distribution by Native Country", x = "Native Country", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Most of the data is from the United States.

Before model creation, let’s make all features factors.

```
data <- data %>%
  mutate_if(is.character, as.factor)
```

Create the Training and Test Sets

Now, it’s time to set aside a validation set comprising 10% of the Adult Census Income dataset. We’ll use “income_training” for training and developing models, as well as for selecting the most effective algorithm. The “final_holdout_test” will then be utilized to evaluate the accuracy of the finalized algorithm.

```
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = data$income, times = 1, p = 0.9, list = FALSE)
income_training <- data[test_index, ]
final_holdout_set <- data[-test_index, ]
```

Modelling Approaches

Let’s further partitioning incomes_data into training and testing sets to test our models, keeping our final holdout set untouched.

```
set.seed(1, sample.kind = "Rounding")
partition <- createDataPartition(y = income_training$income, p = 0.9, list = FALSE)
training <- income_training[partition, ]
testing <- income_training[-partition, ]
```

```
str(training)
```

```
## tibble [26,376 x 13] (S3: tbl_df/tbl/data.frame)
##  $ age           : num [1:26376] 50 38 53 52 31 32 40 25 32 38 ...
##  $ working_class : Factor w/ 9 levels "Federal-gov",..: 7 5 5 7 5 5 5 7 5 5 ...
##  $ education_num : num [1:26376] 13 9 7 9 14 12 11 9 9 7 ...
##  $ marital_status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 3 1 3 3 5 5 3 5 5 3 ...
##  $ occupation    : Factor w/ 15 levels "Adm-clerical",..: 4 6 6 4 11 13 3 5 7 13 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 1 2 1 1 2 2 1 4 5 1 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 5 5 3 2 5 5 5 ...
##  $ gender        : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 2 2 2 2 ...
##  $ capital_gain  : num [1:26376] 0 0 0 0 14084 ...
##  $ capital_loss  : num [1:26376] 0 0 0 0 0 0 0 0 0 0 ...
##  $ hours_per_week: num [1:26376] 13 40 40 45 50 50 40 35 40 50 ...
##  $ native_country: Factor w/ 42 levels "Cambodia","Canada",..: 40 40 40 40 40 40 28 40 40 40 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 2 2 1 2 1 1 1 ...
```

```
str(testing)
```

```
## tibble [2,929 x 13] (S3: tbl_df/tbl/data.frame)
##  $ age           : num [1:2929] 39 28 23 34 49 23 47 46 53 17 ...
##  $ working_class : Factor w/ 9 levels "Federal-gov",..: 8 5 5 5 5 2 5 5 5 4 ...
##  $ education_num : num [1:2929] 13 13 13 4 9 12 15 9 9 6 ...
##  $ marital_status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 5 3 3 5 3 3 1 5 ...
##  $ occupation    : Factor w/ 15 levels "Adm-clerical",..: 1 11 1 15 3 12 11 9 13 8 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 6 4 1 1 2 6 6 4 4 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 3 5 1 5 5 5 5 5 5 ...
##  $ gender        : Factor w/ 2 levels "Female","Male": 2 1 1 2 2 2 1 1 1 1 ...
##  $ capital_gain  : num [1:2929] 2174 0 0 0 0 ...
##  $ capital_loss  : num [1:2929] 0 0 0 0 0 ...
##  $ hours_per_week: num [1:2929] 40 40 30 45 40 52 60 40 35 32 ...
##  $ native_country: Factor w/ 42 levels "Cambodia","Canada",..: 40 5 40 26 40 40 16 40 40 40 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 2 1 1 1 ...
```

Check dimensions to confirm the size of each set.

```
dim(training)
```

```
## [1] 26376    13
```

```
dim(testing)
```

```
## [1] 2929   13
```

Then, create a table to record model performances. This is crucial for comparing models quantitatively and making informed decisions about which model performs best in predicting income levels based on the features provided.

```
model_performances <- data.frame(
  Model = character(),
  Accuracy = numeric(),
  stringsAsFactors = FALSE
)
```

We use accuracy as the primary metric as it provides a straightforward metric for comparing the efficacy of different models.

First Prediction Approach: Logistic Regression

Logistic regression is a robust statistical method for predicting a binary outcome. It's particularly useful for cases where you want to understand the influence of several independent variables on a binary outcome. In our case, it's whether an individual earns more than $50,000 annually.

```
train_control <- trainControl(method = "cv", number = 10, savePredictions = "final")

logistic_model <- train(income~., data=training, method="glm", family="binomial", trControl=train_contr

logistic_predictions <- predict(logistic_model, testing)

accuracy_logistic <- confusionMatrix(logistic_predictions, testing$income)$overall["Accuracy"]

model_performances <- rbind(model_performances, data.frame(Model = "Logistic Regression", Accuracy = ac

print(model_performances)
```

```
##                         Model Accuracy
## Accuracy Logistic Regression 0.852168
```

Logistic regression performed with an accuracy of 85.22%. That is great for our first approach!

Second Prediction Approach: Random Forest

Random Forest is an ensemble learning method known for high accuracy and robustness, particularly effective for datasets with a high dimensionality and a mix of numeric and categorical variables. It builds multiple decision trees and merges them together to get a more accurate and stable prediction.

```
random_forest <- randomForest(income~., data=training, ntree = 500, mtry = 3, importance = TRUE)

accuracy_rf <- confusionMatrix(predict(random_forest, testing), testing$income)$overall["Accuracy"]

model_performances <- rbind(model_performances, data.frame(Model = "Random Forest", Accuracy = accuracy

print(model_performances)
```

```
##                           Model  Accuracy
## Accuracy  Logistic Regression 0.8521680
## Accuracy1       Random Forest 0.8675316
```

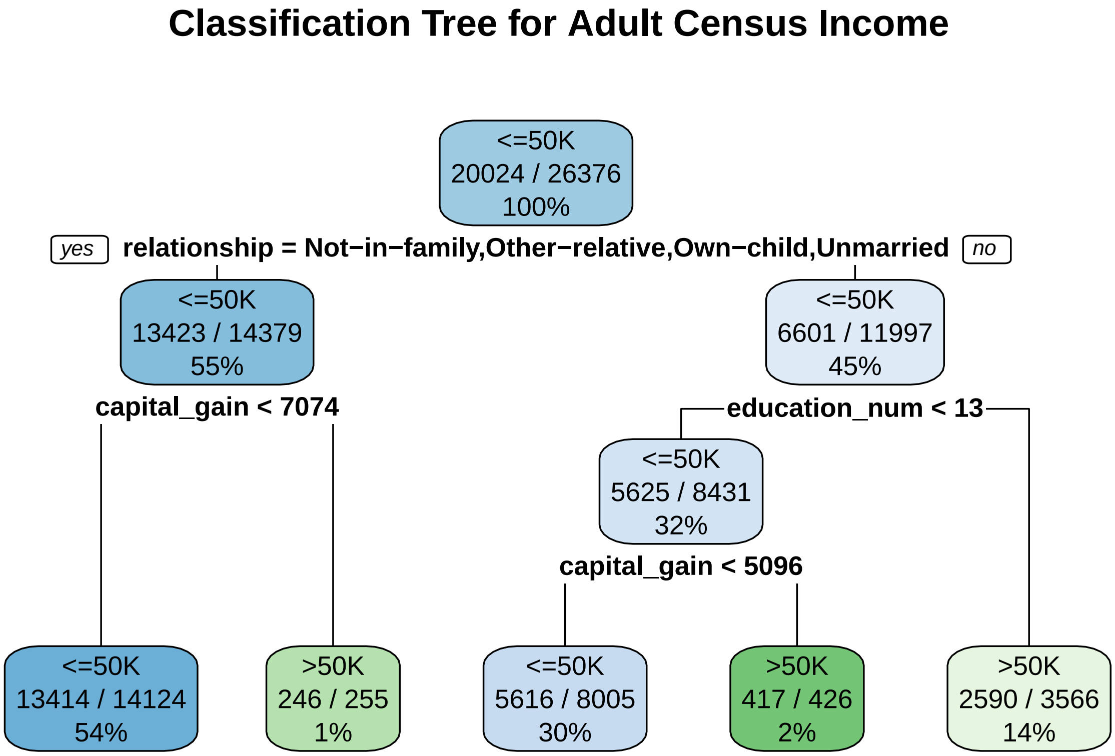
Random Forest had a higher accuracy at 86.75%. Let's try one more to see what we get…

Third Prediction Approach: Classification Tree Model

A decision tree is a simple, interpretable modeling technique. Trees split the data into branches to form a tree structure, making decisions easy to visualize and understand.

```
classification_tree <- rpart(income~., data=training, method="class")

rpart.plot(classification_tree, main="Classification Tree for Adult Census Income", extra=102)
```



```
tree_predictions <- predict(classification_tree, testing, type="class")

Accuracy_tree <- confusionMatrix(tree_predictions, testing$income)$overall["Accuracy"]

model_performances <- rbind(model_performances, data.frame(Model = "Classification Tree", Accuracy = Ac

print(model_performances)
```

```
##                           Model  Accuracy
## Accuracy  Logistic Regression 0.8521680
## Accuracy1       Random Forest 0.8675316
## Accuracy2 Classification Tree 0.8443155
```

Our final approach gave us an accuracy of 84.43%. This means that Random Forest is our highest performing model.

Results

Since Random Forest was our best model, we're going to use this method to perform our final evaluation. This will demonstrate the model's general capability and effectiveness in practical scenarios.

```
final_random_forest <- randomForest(income~., data=income_training, ntree = 500, mtry = 3, importance =

accuracy_final_rf <- confusionMatrix(predict(final_random_forest, final_holdout_set), final_holdout_set

model_performances <- rbind(model_performances, data.frame(Model = "Final Random Forest", Accuracy = ac

print(model_performances)
```

```
##                           Model  Accuracy
## Accuracy  Logistic Regression 0.8521680
## Accuracy1       Random Forest 0.8675316
## Accuracy2 Classification Tree 0.8443155
## Accuracy3 Final Random Forest 0.8694717
```

The final evaluation achieves an accuracy of 86.95%.

Conclusion

The final model demonstrates a significant predictive capability, highlighting the effectiveness of the Random Forest approach in handling complex, multi-dimensional data like the Adult Census Income dataset (with an accuracy of 86.95%). Comparative analysis with Logistic Regression and Classification Tree models indicated that the ensemble method provided a more accurate and stable performance across diverse data subsets. The model's success is promising for applications in socio-economic research and policy making, where accurate income predictions can assist in targeted social programs and resource allocation. Future work can explore other sophisticated ensemble techniques (such as K-Nearest Neighbors) and deep learning models to further enhance predictive accuracy. Additional feature engineering and data augmentation strategies can also be considered to address class imbalance and potential biases in model training and predictions.

References

Irizarry, R. A. (2019). Introduction to data science: Data analysis and prediction algorithms with R. HarvardX. https://leanpub.com/datasciencebook

Kohavi, R., & Becker, B. (1996). UCI Machine Learning Repository: Adult dataset [Data set]. University of California, Irvine, School of Information; Computer Sciences. https://archive.ics.uci.edu/ml/datasets/adult

RStudio. (2024, April 24). Data visualization with ggplot2 [Cheatsheet]. https://learninginnovation.duke.edu/wp-content/uploads/2020/07/R_ggplot2_cheatsheet.pdf