Socioeconomic Profiling Through Census Data Analysis

Final Report

Sushma Reddy Vutla

BUAN 6356 BUSINESS ANALYSIS WITH R | PROFESSOR ZHE ZHANG | DECEMBER 7, 2023
THE UNIVERSITY OF TEXAS AT DALLAS

Executive Summary

This document provides a summary of the key information and steps involved in the project titled "Socioeconomic Profiling through Census Data Analysis." The project aims to analyze census data from the 1994 Census database to create a socioeconomic profile. The primary objectives are income prediction and exploring the relationship between income levels and various demographic and employment-related features.

Project Motivation/Background

Understanding socioeconomic factors is crucial for policy-making, resource allocation, and addressing disparities within a population. This project utilizes census data to predict income levels and identify patterns based on factors such as age, education, occupation, marital status, gender, race, and more.

Data Description

Data Source

The dataset used for analysis is the Adult Dataset from the UCI ML repository, extracted from the 1994 Census database.

Data Size and Volume

- Total Instances: 48,842
- Features: 14 (6 numerical, 8 categorical)
- Training Set: 32,561 instances
- Test Set: 16,281 instances

Variables and Features

- Key features include age, workclass, education, marital status, occupation, relationship, race, sex, capital gain, capital loss, hours per week, country, and class.

Data Quality and Cleaning

Missing Attribute Values

- 7% of instances have missing values.
- Unknown instances were removed, resulting in 45,222 instances for analysis.

Class Distribution

- Probability for the label '>50K': 23.93% (24.78% without unknowns).

Data Cleaning and Quality Check (R Code)

The R code used for data cleaning and quality check involved the following steps:

1. Handling Missing Values:
   - Replaced '?' with NA in both training and testing sets.
2. Identifying and Converting Character Columns:
   - Identified common character columns between training and testing sets.
   - Converted common character columns to factors.

Imputing Missing Values:

- Imputed missing values with mean for numerical columns and mode for categorical columns.

Snapshot

Total instances: 45,222

Income distribution: >50K (23.93%), <=50K (76.07%)

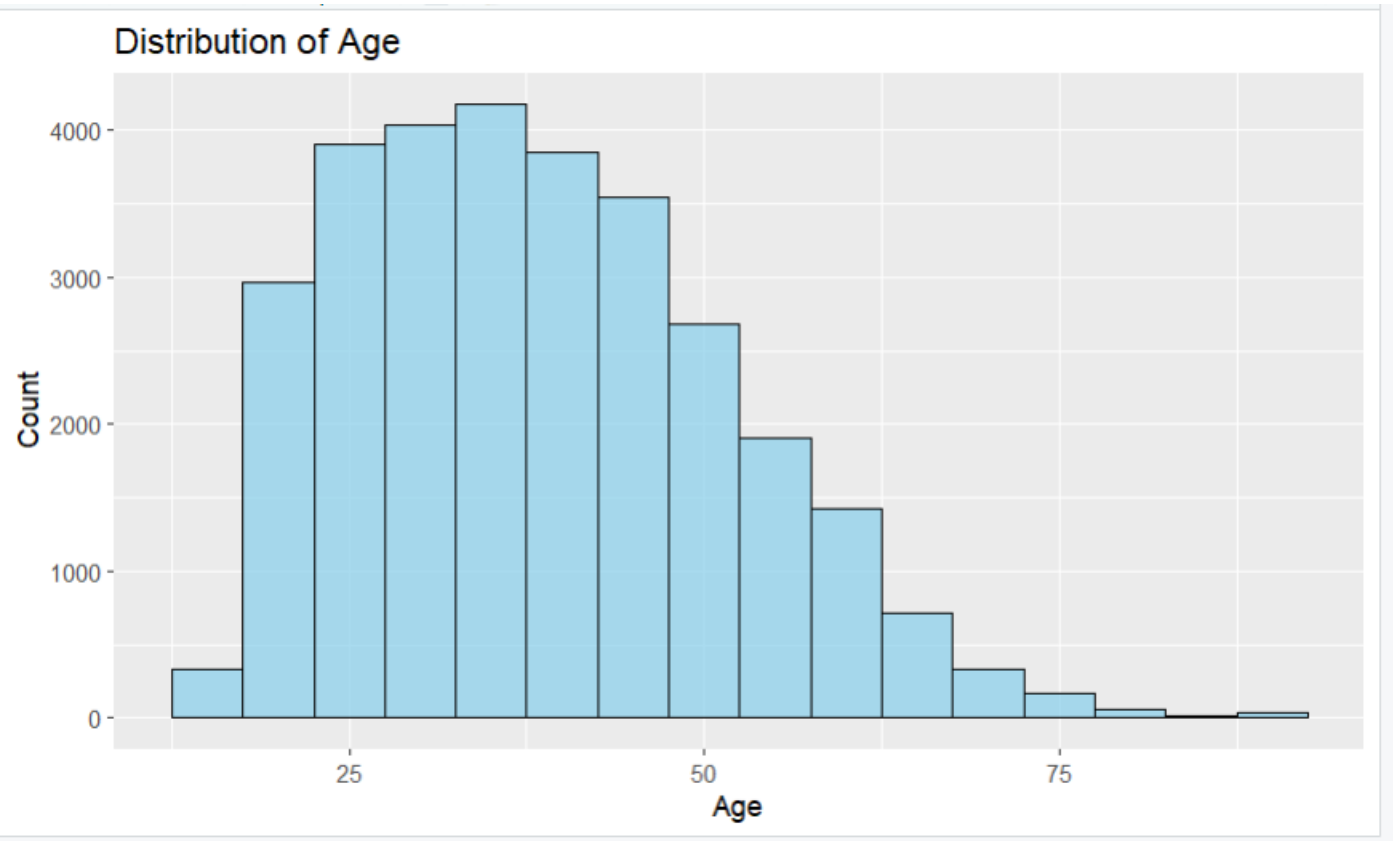
Key demographic factors

Exploratory Data Analysis

Histogram: Distribution of Age

Description:

The histogram visually represents the distribution of ages in the imputed training dataset. The x-axis displays age values, and the y-axis shows the count of individuals within each age bin. The binwidth is set to 5 to capture the granularity of age groups.



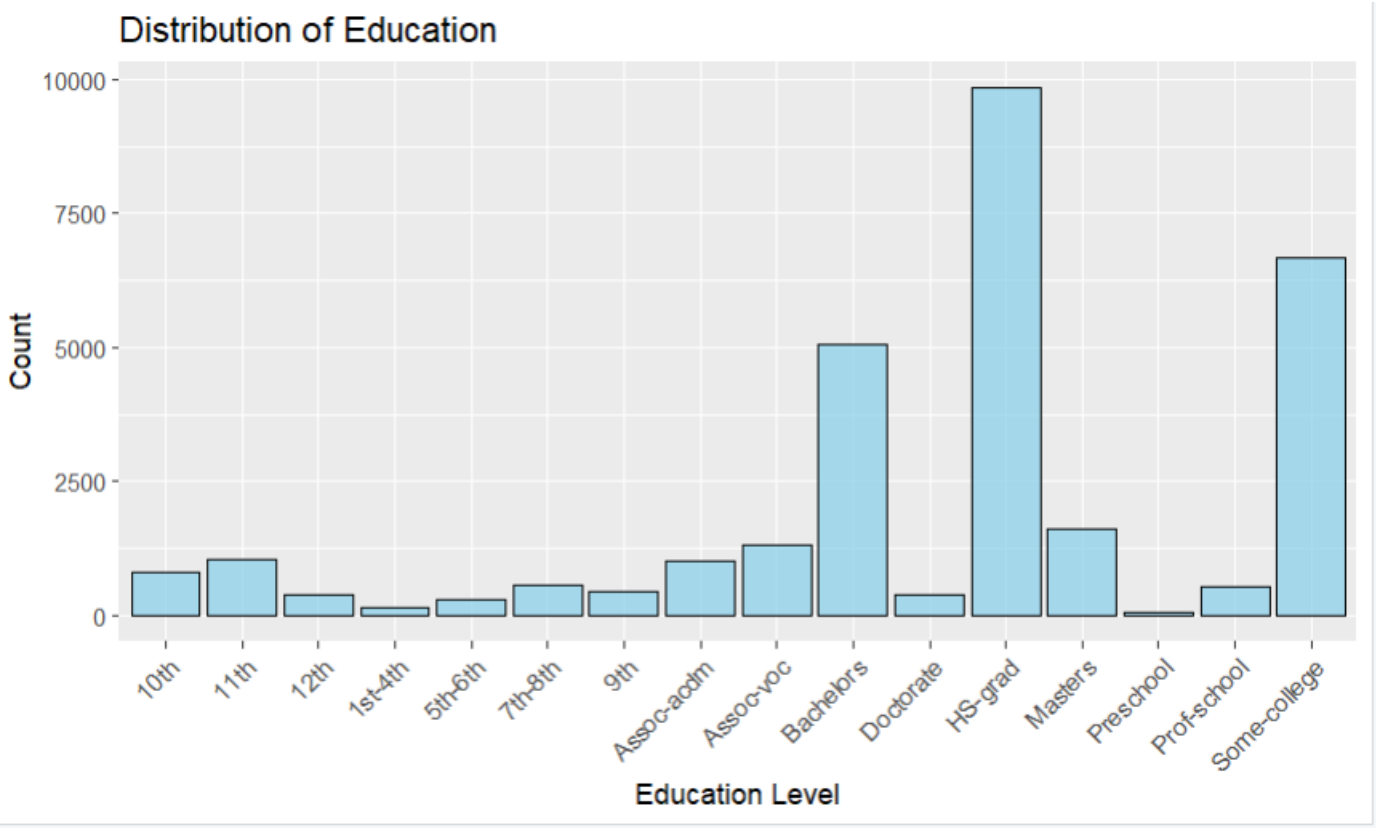
Insights:

The age distribution reveals that the dataset is skewed towards individuals in their 20s to 50s, with a peak around the age of 30. Understanding the age demographics is crucial for the income prediction model as certain age groups may exhibit distinct income patterns.

Histogram: Distribution of Education

Description:

This histogram displays the distribution of education levels in the imputed training dataset. The x-axis represents different education categories, and the y-axis shows the count of individuals within each category.



Insights:

The distribution highlights that a significant portion of individuals in the dataset has completed education up to the "HS-grad" level. Education is a pivotal factor in income prediction, as individuals with higher educational attainment may have distinct income profiles.

Correlation Analysis: Numerical Features

Correlation Matrix

```
> print(correlation_matrix)
                     age hours-per-week capital-gain capital-loss     class
age           1.00000000     0.10159876   0.08015423   0.06016548 0.2419981
hours-per-week 0.10159876     1.00000000   0.08043180   0.05241705 0.2294801
capital-gain   0.08015423     0.08043180   1.00000000  -0.03222933 0.2211962
capital-loss   0.06016548     0.05241705  -0.03222933   1.00000000 0.1500533
class          0.24199814     0.22948013   0.22119621   0.15005331 1.0000000
```

Key Insights:

Age vs. Class:

Positive correlation (0.24) between age and income class. Older individuals tend to have higher income levels.

Hours per Week vs. Class:

Moderate positive correlation (0.23) between hours worked per week and income class. Longer working hours may correlate with higher income.

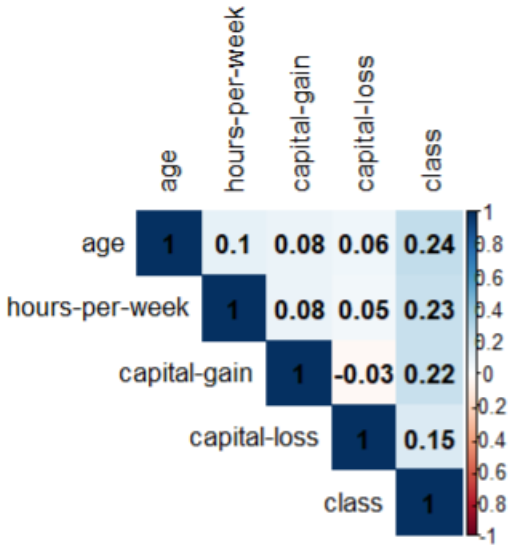
Capital Gain vs. Class:

Positive correlation (0.22) between capital gain and income class. Individuals with higher capital gains are more likely to be in the ">50K" income class.

Capital Loss vs. Class:

Positive correlation (0.15) between capital loss and income class. The impact of capital loss on income class is weaker compared to other factors.

Heatmap Visualization:



Correlation Heatmap

The heatmap visually represents the correlation matrix. Darker colors indicate stronger correlations. This visualization enhances the understanding of relationships between numerical features and income levels.

Conclusion:

The correlation matrix and heatmap provide comprehensive insights into the relationships between numerical features and income levels. These findings will guide the development of a predictive model for income classification

Meeting Data Mining Objectives

These exploratory analyses contribute to the Income Prediction objective by providing a foundational understanding of key demographic features. Age and education, as revealed by the histograms, showcase patterns that can significantly influence income levels. Identifying these patterns is crucial for developing a robust predictive model and can guide feature selection and engineering in subsequent modeling stages.

Conclusion

The project lays the foundation for socioeconomic profiling based on census data. The cleaned and imputed dataset is ready for further analysis and model development to predict income levels and uncover insights into the relationship between income and demographic factors.

Data Preprocessing

- **Remove unnecessary columns:**

Removing unnecessary columns is part of the data preprocessing phase, and it serves several purposes, especially when working with tree-based models like decision trees. Reasons for removing unnecessary columns is important:

1. Reduced Dimensionality:

Removing unnecessary columns reduces the dimensionality of the dataset. High-dimensional datasets can lead to increased computational complexity and longer training times, especially for tree-based models.

2. Improved Model Performance:

Unnecessary columns might introduce noise to the model or lead to overfitting. By removing irrelevant features, you focus the model on the most relevant information, potentially improving its generalization to new, unseen data.

3. Simpler Model Interpretation:

Decision trees can become complex and difficult to interpret when dealing with many features. By removing irrelevant columns, the resulting tree will be simpler and more interpretable, making it easier to understand the decision-making process.

4. Faster Training Time:

Tree-based models, such as decision trees and random forests, build a tree structure based on the features. The fewer features there are, the faster the model can be trained. This is particularly beneficial when working with large datasets.

- Eliminated columns with indices fnlwgt, education, relationship, marital_status, capital gain, and capital_loss and from both training and validation datasets.

```
# Read the data
testing.data <- read.csv("testing.data.csv")
validation.data <- read.csv("training.data.csv")
# Remove any unnecessary columns
train.data <- testing.data[, -c(3, 4, 6, 8, 11, 12)]
valid.data <- validation.data[, -c(3, 4, 6, 8, 11, 12)]
```

- **Encode categorical variables:**

Encoding categorical variables is a crucial step in the data preprocessing pipeline, especially when working with machine learning models. Categorical variables represent qualitative data and can take on values from a limited, fixed set. Machine learning algorithms typically require numerical input, so encoding is necessary to convert categorical variables into a format that can be used for model training.

Decision tree-based models, like Random Forest and Gradient Boosting, can handle categorical variables without explicit encoding. However, explicit encoding might still be beneficial for consistency and ease of interpretation.

- Converted categorical variables like 'workclass,' 'occupation,' class, 'race,' 'sex,' and 'country' into numerical format using techniques like one-hot encoding.

```
# Convert categorical variables to factors
train.data$workclass <- as.factor(train.data$workclass)
train.data$occupation <- as.factor(train.data$occupation)
train.data$race <- as.factor(train.data$race)
train.data$sex <- as.factor(train.data$sex)
train.data$country <- as.factor(train.data$country)
train.data$class <- as.factor(train.data$class)
summary(train.data)
# Convert categorical variables to factors
valid.data$workclass <- as.factor(valid.data$workclass)
valid.data$occupation <- as.factor(valid.data$occupation)
valid.data$race <- as.factor(valid.data$race)
valid.data$sex <- as.factor(valid.data$sex)
valid.data$country <- as.factor(valid.data$country)
valid.data$class <- as.factor(valid.data$class)
summary(valid.data)
```

Decision Tree Model

Building a decision tree model for income prediction using the Adult Dataset

➢ Model Training: Building the Decision Tree Model

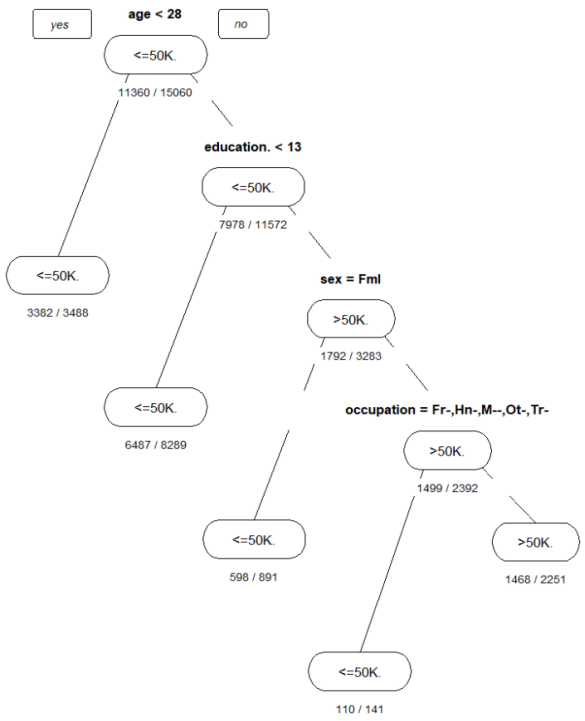
Defined features (X) and target variable (y), where X includes all the features, and y is the income class ('>50K' or '<=50K').

The criteria for making decisions at each node is determined based on impurity measure information gain.

```
# classification tree split using entropy
default.ct <- rpart(class ~ ., data = train.data, parms = list(split = 'information'), method = "class")
prp(default.ct, type = 1, extra = 2, under = TRUE, split.font = 2, varlen = -10)
length(default.ct$frame$var[default.ct$frame$var == "<leaf>"])
```

- ❑ No of Leaves: 5
- ❑ Split based on:
  - Age
  - Workclass
  - Education_level
  - Occupation

Visualize the Decision Tree



➢ Model Evaluation for Training data

Evaluate the performance of model using metrics like accuracy, sensitivity, specificity, and confusion matrix to understand how well the model is predicting each income class.

```
# classify records in the TRAINING data.
```

```
# set argument type = "class" in predict() to generate predicted class membership.
pred.train <- predict(default.ct, train.data,type = "class")
# generate confusion matrix for training data
confusionMatrix(pred.train, as.factor(train.data$class))
Confusion Matrix and Statistics

          Reference
Prediction <=50K. >50K.
     <=50K.  10577  2232
     >50K.     783  1468

               Accuracy : 0.7998
                 95% CI : (0.7933, 0.8062)
    No Information Rate : 0.7543
    P-Value [Acc > NIR] : < 2.2e-16
                  Kappa : 0.3777
 Mcnemar's Test P-Value : < 2.2e-16
            Sensitivity : 0.9311
            Specificity : 0.3968
         Pos Pred Value : 0.8257
         Neg Pred Value : 0.6522
             Prevalence : 0.7543
         Detection Rate : 0.7023
   Detection Prevalence : 0.8505
      Balanced Accuracy : 0.6639

       'Positive' Class : <=50K.
```

➢ Interpretation:

Interpreting to understand the most important features that contribute to predicting income levels.

Interpreting the confusion matrix and statistics for the training model:

- True Positive (TP): 10577
- False Positive (FP): 783
- True Negative (TN): 1468
- False Negative (FN):2232

Statistics:

- **Accuracy:** 79.98%
  - The proportion of correctly classified instances.
- **95% Confidence Interval (CI):** (79.33%, 80.62%)
  - A range that provides an estimated probability that the true accuracy of the model lies within this interval.
- **Sensitivity (True Positive Rate):** 93.11%

- The proportion of true positives among all actual positive instances.

- **Specificity (True Negative Rate):** 39.68%

- The proportion of true negatives among all actual negative instances.

- **Positive Predictive Value (Precision):** 82.57%

- The proportion of true positives among all predicted positives.

- **Negative Predictive Value:** 65.22%

- The proportion of true negatives among all predicted negatives.

- The average of sensitivity and specificity, providing a balanced view of classification performance.
- The model shows relatively high accuracy (79.98%)
- Sensitivity is high (93.11%), indicating that the model is good at identifying individuals with income less than or equal to $50K.
- Specificity is relatively low (39.68%), suggesting that the model is less accurate in identifying individuals with income greater than $50K.
- Positive Predictive Value (Precision) is decent at 82.57%, indicating that when the model predicts '>50K', it is correct about 82.57% of the time.

➢ Model Evaluation for Validation data

Evaluate the performance of your model using metrics like accuracy, sensitivity, specificity and confusion matrix to understand how well the model is predicting each income class.

Make Predictions on validation Data:

```
# classify records in the VALIDATION data.
# set argument type = "class" in predict() to generate predicted class membership.
preds.train <- predict(default.ct, valid.data,type = "class")
# Check and set levels to be the same
levels(preds.train) <- levels(valid.data$class)
# proceed with confusionMatrix
 confusionMatrix(valid.data$class, as.factor(preds.train))
Confusion Matrix and Statistics

                Reference
Prediction <=50K  >50K
     <=50K 21127  1526
     >50K   4517  2991

               Accuracy : 0.7996
                 95% CI : (0.7951, 0.8041)
    No Information Rate : 0.8502
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3819

 Mcnemar's Test P-Value : <2e-16
```

Sensitivity : 0.8239
Specificity : 0.6622
Pos Pred Value : 0.9326
Neg Pred Value : 0.3984
Prevalence : 0.8502
Detection Rate : 0.7005
Detection Prevalence : 0.7511
Balanced Accuracy : 0.7430

'Positive' Class : <=50K

- **Interpretation:**

Interpreting the confusion matrix and statistics for the validation model:

- True Positive (TP): 21127

- False Positive (FP):4517

- Negative (TN): 2991

- False Negative (FN): 1526

Statistics:

- **Accuracy:** 79.96%

  - The proportion of correctly classified instances.

- **95% Confidence Interval (CI):** (79.51%, 80.41%)

  - A range that provides an estimated probability that the true accuracy of the model lies within this interval.

- **Sensitivity (True Positive Rate):** 82.39%

  - The proportion of true positives among all actual positive instances.

- **Specificity (True Negative Rate):** 66.22%

  - The proportion of true negatives among all actual negative instances.

- **Positive Predictive Value (Precision):** 93.26%

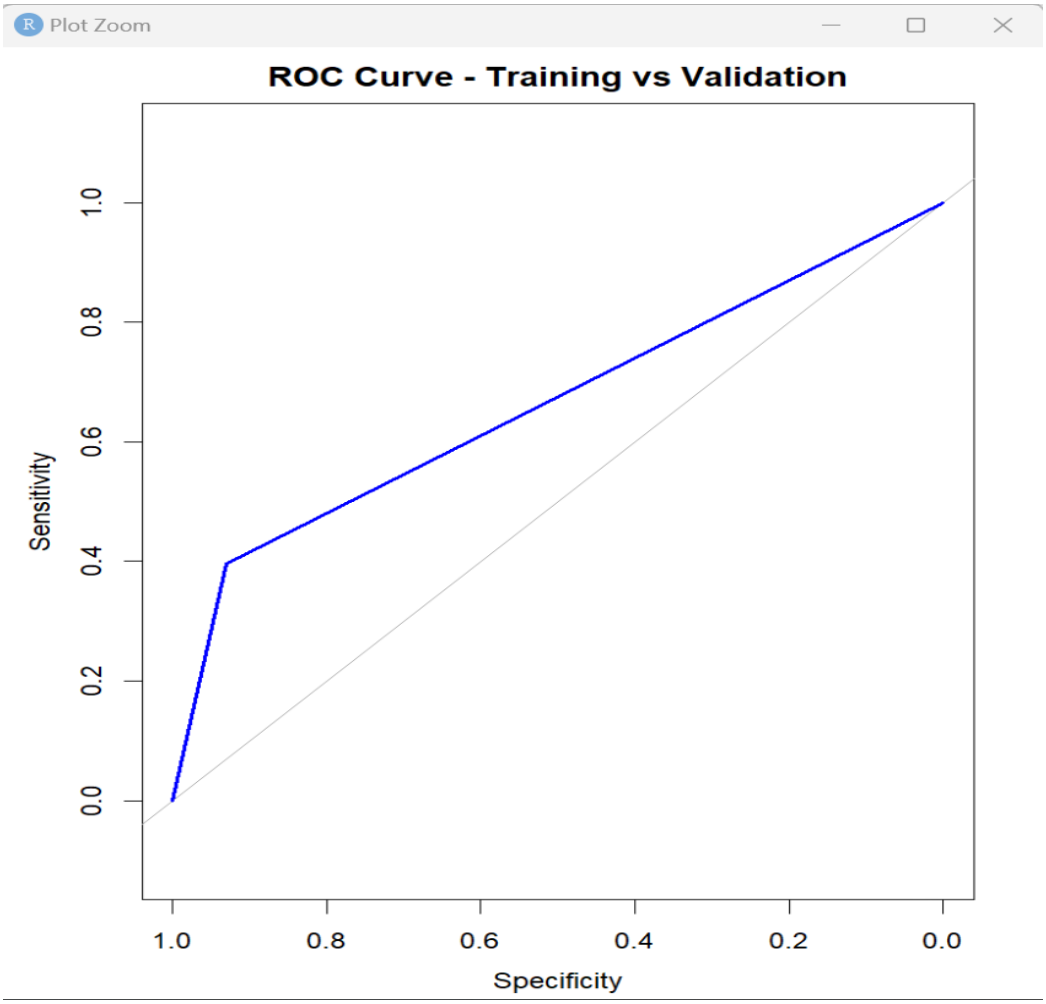
  - The proportion of true positives among all predicted positives.

- **Negative Predictive Value:** 39.84%

  - The proportion of true negatives among all predicted negatives.

- The model has an overall accuracy of 79.96%, performing reasonably well in classifying both income groups.
- Sensitivity is 82.39%, indicating that the model is good at identifying individuals with income less than or equal to $50K.
- Specificity is 66.22%, suggesting that the model is moderately accurate in identifying individuals with income greater than $50K.
- Positive Predictive Value (Precision) is high at 93.26%, indicating that when the model predicts '>50K', it is correct about 93.26% of the time.

Comparison and Interpretation of Training and validation data



- **Area under the curve: 0.7639**

Accuracy:

Training: 79.98%

Validation: 79.96%

Both datasets show similar accuracy, indicating that the model performs consistently on both training and unseen data.

Sensitivity (True Positive Rate):

Training: 93.11%

Validation: 82.39%

Sensitivity is the proportion of actual positive instances correctly predicted as positive. It's slightly higher in the training set, suggesting the model may perform slightly better on the training data.

Specificity (True Negative Rate):

Training: 39.68%

Validation: 66.22%

Specificity is the proportion of actual negative instances correctly predicted as negative. It is notably higher in the validation set, suggesting the model is better at avoiding false positives in unseen data.

Positive Predictive Value (Precision):

Training: 82.57%

Validation: 93.26%

Precision is the proportion of predicted positives that are true positives. It's slightly higher in the validation set, indicating fewer false positives in predictions on the validation data.

In summary, while the model's performance is consistent between training and validation, there are some differences in sensitivity, specificity, and balanced accuracy.