Analysis of the Census Income dataset

Joel Solé and Carlos Arbonés

GCED, UPC.

Abstract

This paper describes the use of the Census Income Data Set, obtained from the UC Irvine Machine Learning Repository, to develop a machine learning model that predicts whether a person's annual income is above or below $50,000 based on 42 explanatory variables. The goal of this study is to explore the relationships between the demographic and employment-related characteristics of individuals and their income levels, and to identify the most important predictors of income. The data set includes weighted census data extracted from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau, and contains 200k rows and 42 columns. The paper outlines the steps taken to preprocess the data, select and tune the model, evaluate its performance, and interpret the importance of each explanatory variable. The results of this study provide insights into the factors that are associated with higher or lower incomes, and highlight the potential biases or limitations of the model. This study contributes to the broader field of data science and machine learning, and has practical implications for policy makers, employers, and individuals who are interested in understanding the determinants of income inequality.

Contents

1 Related previous work

2 Data exploration

In this section, we will explore the Census-Income dataset, which contains information about individuals and their income. We begin by reading the data and dropping the ”unknown” column, as it should not be used for classifiers according to the dataset description.

2.1 Basic inspection of the data

The first thing we observe when analyzing this dataset is that out of the 41 variables it contains, 29 are categorical and 12 are numeric (integer type). Additionally, we have a few numerical variables that should actually be treated as categorical, such as ”det_ind_code” or ”det_occ_code” that represent detailed industry and occupation codes respectively, as well as age and year, which we should treat as categorical variables.

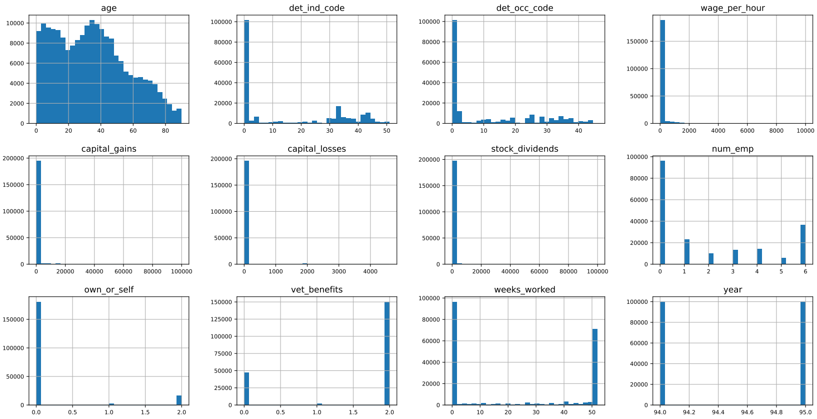
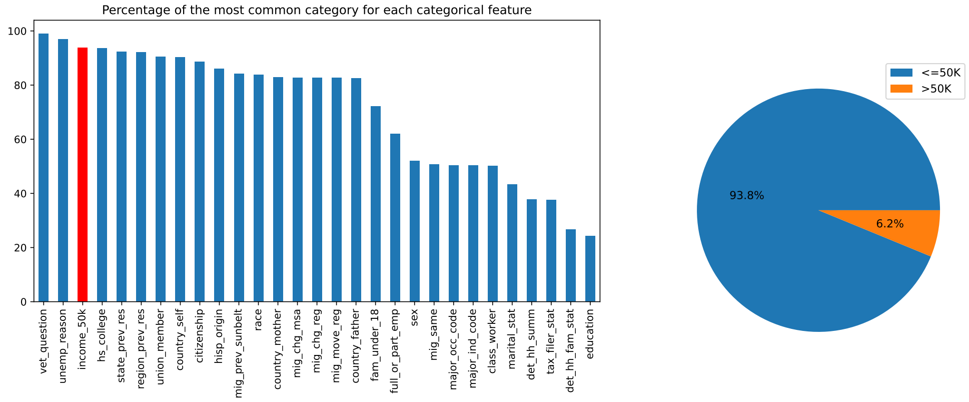


Fig. 1: Histogram for all the numerical data

Another thing we observe about the numerical variables is that, apart from age and year (which we have already mentioned are not really categorical), the rest have the vast majority of values at 0. This imbalance present in the numerical variables makes it difficult to fit some models, so we to fit models that can work well with categorical variables, such as logistic regression, decision trees, random forests, and support vector machines.

On the other hand, with the Figure 2a we observe that the categorical data is also unbalanced, as more than half of the features have a category that contains more than 80% of the data. As we can see in Figure 2b, we also have a significant imbalance in our target variable. This can lead to the model being biased towards the majority class, making it difficult to accurately predict the minority class. For example, if a classification model is trained on an imbalanced dataset where 90% of the samples belong to class A and 10% belong to class B, the model may achieve high accuracy by simply predicting all samples as class A. This is because the model has not seen enough examples of class B to learn how to distinguish it from class A. In addition, the evaluation metrics used to measure the performance of the model, such as accuracy, can be misleading in



(a) Percentage of the most common category for each categorical feature (target value plotted in red)

(b) Frequencies of the target variable value

Fig. 2: Categorical data analysis

imbalanced datasets. Therefore, it is important to address the class imbalance issue by using techniques such as oversampling, undersampling, or using models that can work well with unbalanced data.

2.2 Data cleaning

As we have seen during the data inspection, we have two options to address the strong imbalance in the data: either apply oversampling or undersampling techniques to artificially balance the dataset, or use the imbalanced dataset and adjust models that can handle strong imbalance.

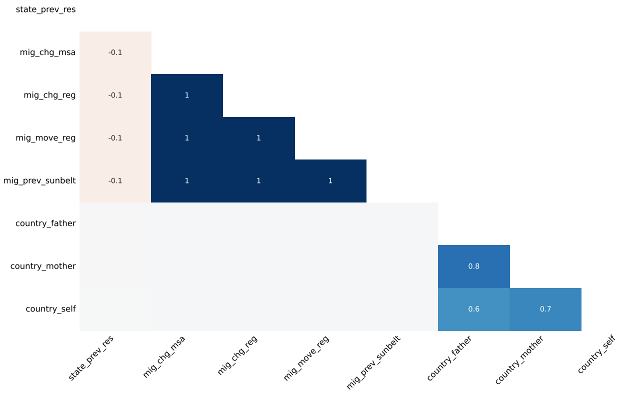
Therefore, we will generate two different datasets that follow the two options mentioned above. Since most of the algorithms that work with categorical data only perform well with an imbalanced dataset, such as decision trees, random forests, or support vector machines, we will maintain an imbalanced dataset with only categorical data. On the other hand, we will generate a balanced dataset using undersampling techniques to fit models that do require a balanced dataset, such as logistic regression.

As a first step, we have removed duplicates from the dataset to ensure data cleanliness and consistency. Having duplicate data in a dataset can lead to biased and inaccurate results. This is because duplicate data can skew the distribution of values and lead to over-representation of certain data points. Additionally, it can result in inefficiencies in data processing and model training, as the same information is being analyzed multiple times.

2.2.1 Missing values treatment

In this section, we will examine the occurrence of missing values within the dataset and outline the appropriate strategies for addressing each case. The handling of missing data is a crucial step in the data preprocessing phase, as it can have a significant impact on the quality and reliability of the analysis and modeling results. Therefore, it is essential to carefully evaluate and handle missing data appropriately to avoid any biases or errors in the downstream analyses.

In the figure 3, we observe that we have 4 columns that have 49% of missing values: mig_chg_msa, mig_chg_reg, mig_move_reg, and mig_prev_sunbelt, and that the correlation between these 4 features is 1. This means that every time we have a missing value



Column	Missing values
state_prev_res	0.46%
mig_chg_msa	49.02%
mig_chg_reg	49.02%
mig_move_reg	49.02%
mig_prev_sunbelt	49.02%
country_father	4.17%
country_mother	3.80%
country_self	2.17%

Fig. 3: Distribution of the missing values

in any of these columns, the others will also have missing values. It makes sense for these variables to be correlated since the 4 columns contain different information about migration (we can see a brief description of these columns in Table 1).

On the other hand, we have that the missing values in country_father, country_mother, and country_self also have a certain correlation, although it is not as high as the 4 previous features.

For the correlated missing values, such as mig_chg_msa, mig_chg_reg, mig_move_reg, and min_prev_sunbelt, we have removed the entire columns since they have a high percentage of missing values (49%), and therefore do not have enough meaningful data to be considered significant. Additionally, imputation does not make sense in this case due to the large number of missing values.

For the other variables with missing values, which account for less than 5% of the data in all cases, we will perform imputation using the most frequent value of the respective variable.

2.2.2 Categorical conversion

We will begin by transforming numeric variables that are in fact categorical, such as det_ind_code and det_occ_code, which respectively represent industry and occupation codes, and also own_or_self, vet_benefits, and year, which can be directly converted into categorical variables since they have between 2 and 3 categories only.

To convert age into a categorical variable, we generate 10 bins: from 0 to 18, from 18 to 25, and in 10-year intervals up to 105. We can observe the resulting categorical features after we converted them in the figure 7

The remaining numerical features in the dataset can be regarded as numerical variables since they include economic variables such as wage_per_hour, capital_gains, capital_losses, and stock_dividends, as well as count variables such as num_emp and weeks_worked. However, it is worth noting that the remaining numerical variables exhibit a high degree of skewness, as depicted in Figure 4, with a substantial number of instances having the same value. As a result, no transformation method can be applied to normalize them.

We need to take into account the highly imbalanced nature of the data when fitting a logistic regression model, as downsampling of samples may be necessary to achieve a better distribution. However, for random forest and decision tree models, this transformation may not be necessary.

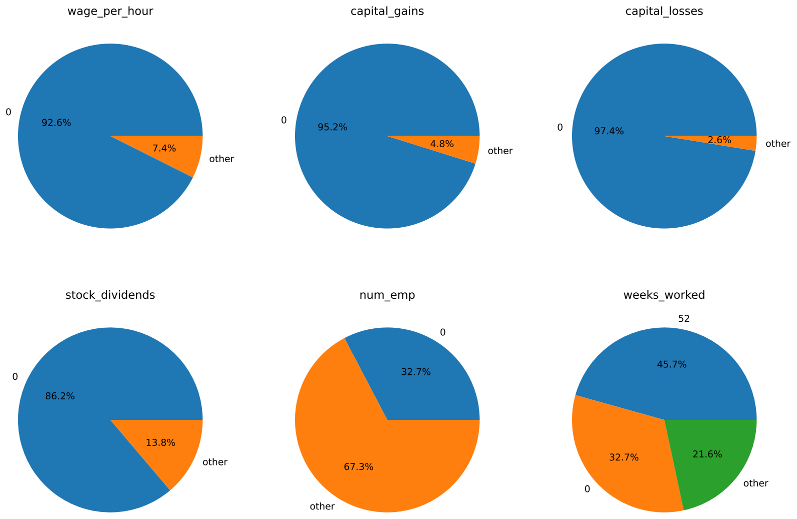


Fig. 4: Most frequent values of the numerical data

3 Model evaluation

In this section, we will evaluate the performance of the applied models after preprocessing. So far, we have applied the following models: decision tree and random forest. Note that after preprocessing, the target variable has a distribution of 91.8% of values with an income below 50K, thus a naive model always predicting ”below 50K” will have a 91.8% accuracy. This is relevant in order to compare the accuracies found after adjusting the models.

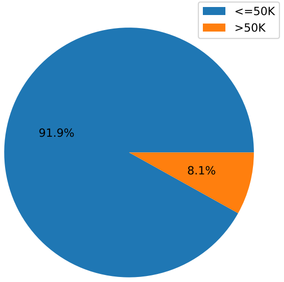
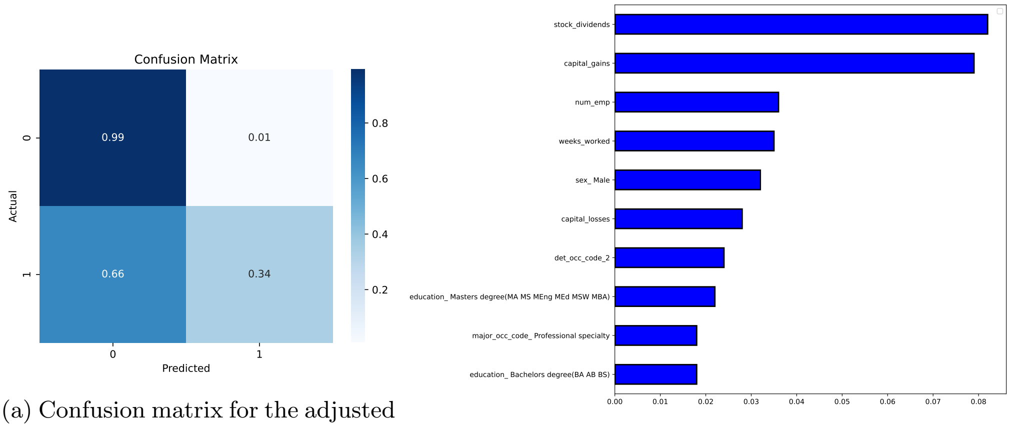


Fig. 5: Distribution of the target variable after data preprocessing

3.1 Random Forest

For the present model, it was necessary to perform a conversion of the categorical variables in the dataset using the one-hot encoding technique. This conversion led to an increase in the dimensionality of the dataset from 40 features to 452 features. The model was implemented using a random forest classifier, with 22 estimators and a maximum depth of 30. The accuracy achieved by the model was 93.8%. Upon examining the confusion matrix presented in figure 6a, it was observed that the model only exhibited errors in cases where it predicted "<=50K" instead of ">50K".



(a) Confusion matrix for the adjusted model

(b) 10 most important features for the adjusted model

Fig. 6: Random forest classifier

Table 1: Column description

Column Name	Description
age	Age of the worker
class_worker	Class of worker
det_ind_code	Industry code
det_occ_code	Occupation code
education	Level of education
wage_per_hour	Wage per hour
hs_college	Enrolled in educational institution last week
marital_stat	Marital status
major_ind_code	Major industry code
major_occ_code	Major occupation code
race	Race
hisp_origin	Hispanic origin
sex	Sex
union_member	Member of a labor union
unemp_reason	Reason for unemployment
full_or_part_emp	Full- or part-time employment status
capital_gains	Capital gains
capital_losses	Capital losses
stock_dividends	Dividends from stocks
tax_filer_stat	Tax filer status
region_prev_res	Region of previous residence
state_prev_res	State of previous residence
det_hh_fam_stat	Detailed household and family status
det_hh_summ	Detailed household summary in household
mig_chg_msa	Migration code - change in MSA
mig_chg_reg	Migration code - change in region
mig_move_reg	Migration code - move within region
mig_same	Live in this house one year ago
mig_prev_sunbelt	Migration - previous residence in sunbelt
num_emp	Number of persons that worked for employer
fam_under_18	Family members under 18
country_father	Country of birth father
country_mother	Country of birth mother
country_self	Country of birth
citizenship	Citizenship
own_or_self	Own business or self-employed?
vet_question	Fill included questionnaire for Veterans Admini...
vet_benefits	Veterans benefits
weeks_worked	Weeks worked in the year
year	Year of survey
income_50k	Income less than or greater than $50,000
edu_year	Number of years of education

4 Appendix

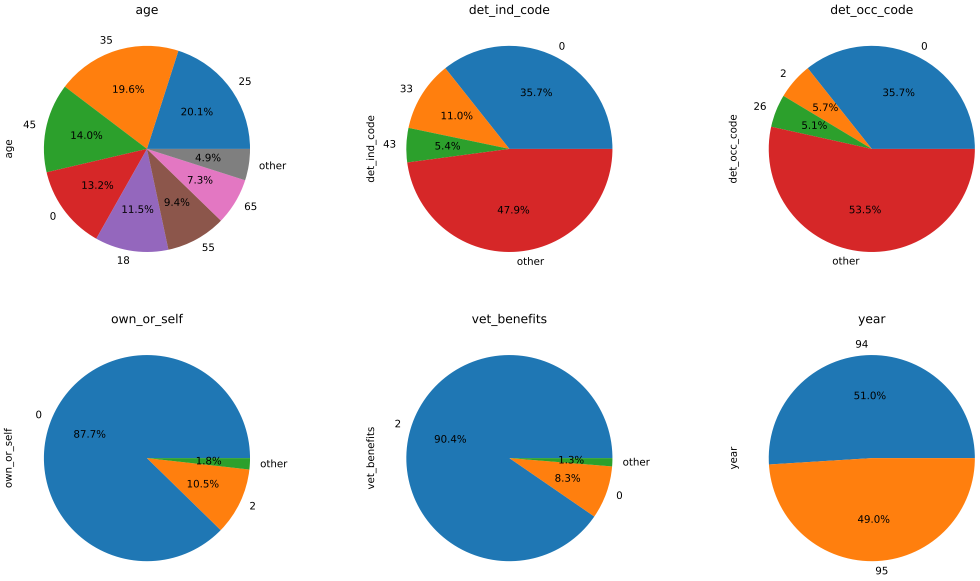


Fig. 7: Resulted categorical data after conversion