Predict whether annual income of an individual exceeds $50K/yr based on census data

Team 101

Gary Zeng

Kienlac Mai

Jacky Cheng

Introduction

For this project, we examine the Adult Census Income dataset available in the UC Irvine Machine Learning Repository, which consists of 48,842 instances extracted from the 1994 Census database. We aim to predict whether a person's income will be greater than $50,000 a year based on several variables defined in the census datatable.

In the Initial Analysis section, we explore various patterns and trends in the census data, and utilize these insights to inform our model selection process. In the Predictive Modeling section, we train and optimize machine learning models to predict individual income, as well as document the performance of alternative variations conceived by tuning parameters to minimize overfitting and underfitting. Lastly, in the Results section, we compare model performance on the test data to identify the most accurate model, key approaches to optimization, and notable features that drive individual income prediction.

Initial Analysis

The Adult Census Income dataset contains 48,842 entries, with each corresponding row containing 14 features. In order to identify the features that were key to addressing the predictive task of this assignment, we initially observed each of the 14 features and their respective distributions of entries that are labeled > 50k and <= 50k.

Label	Number	Percentage

<=50k	24720	75.919
>50k	7841	24.081

We initially observed that about 75.919% of the entries were labeled with <=50k and about 24.081% of the entries are labeled with >50k. We then split the dataset into separate training and test sets while maintaining this distribution, and created the following bar graphs for each feature that we determined to be the most relevant and important based on our initial analysis of the dataset in the context of the given predictive task.

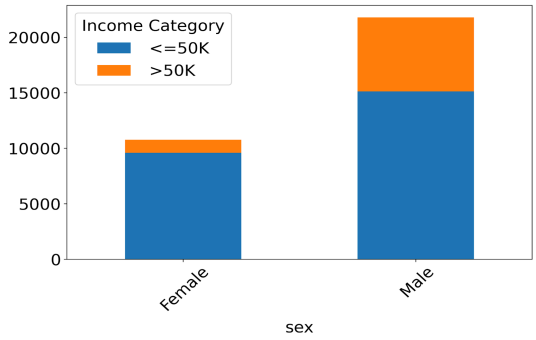

Figure 1. The age range between 40 - 50 has the highest percentages of professionals making more than 50k a year.

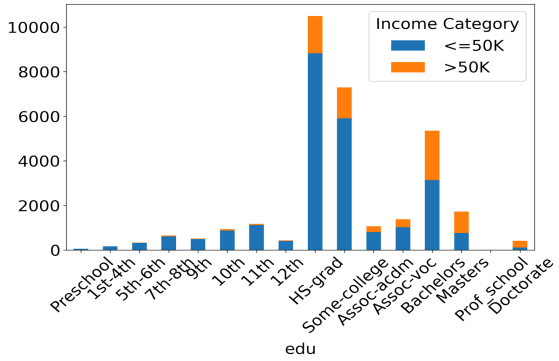

Figure 2. With more people making 50k in higher education, this indicates having a higher education has a higher chance of making above 50k a year.

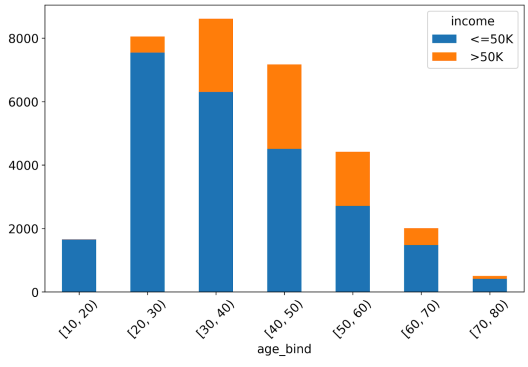

Figure 3. Private Jobs has the highest percentages of population that make more than 50k a year.

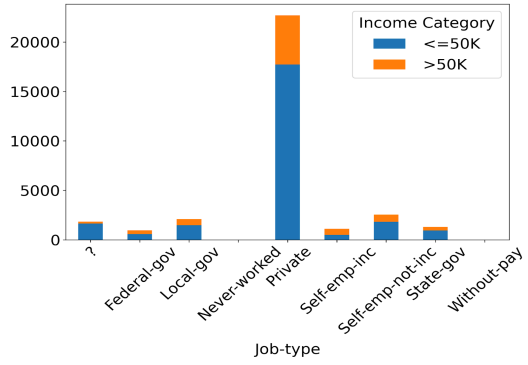

Figure 4. Professional making more than 50k a year this indicates having a higher education has a higher

chance of making above 50k a year.

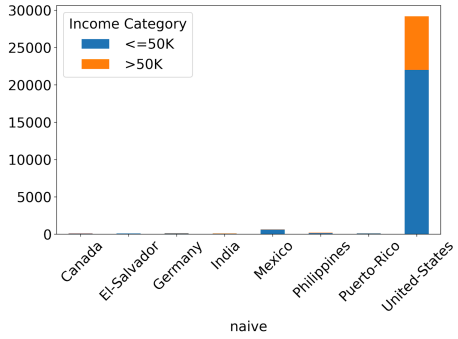

Figure 5. Most people are from the United States. We have to clear some countries to create these graphs, but they are in our models.

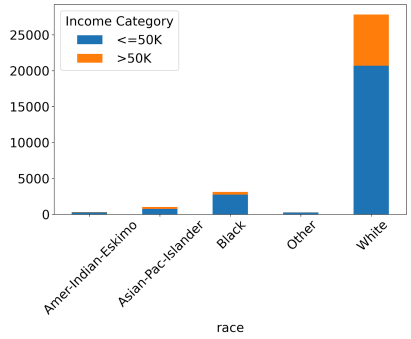

Figure 6. A majority of the dataset population is white. They also have the highest percentages worker that has more than 50k per year.

As seen in the uneven distribution of the graphs, possible biases in data distribution may be present in this data set, which may influence the predictive outcomes of our associated models. All substantial biases may be attributed to the large sample size originating from the United States, where more socioeconomic opportunities are more accessible compared to other countries. Though this factor may not be a necessary substantial consideration when analyzing this dataset, this distribution is worth noting.

Predictive Modeling

Logistic Regression - Kienlac Mai

For our initial model we decided to utilize a Logistic Regression as a predictive representation for both the categorical and the quantitative features provided in the dataset. In order to get a complete sense of all the weights among the data, all 13 features, excluding the target variable, were utilized in the training of the regression model, despite some features seeming more socially influential, such as hours worked per week and educational levels. The

features were processed into numerical and categorical data, and multiple variations of the model were created, with varying strengths of regularization {0.1, 1, 10} employed to alter the complexity of each model. A 5-fold split was utilized on the dataset with 80% categorized as the training data, and the remaining 20% being the testing data. After each model was trained on the testing data, the predictive error rate resulted in 14.89%.

	Training Error	**Test Error**
Logistic Regression	14.89%	14.62%

Naive Bayes - Jacky Cheng

Next, we utilized Naive Bayes models because they are well-suited for classification tasks involving mixed data types, making them an ideal fit for addressing the categorical (e.g. job type, education, marital status) and continuous (e.g. age, hours per week) features of the Adult dataset. We tested three Naive Bayes variants to account for these different data types: Gaussian Naive Bayes for handling continuous features under the assumption of a normal distribution, Multinomial Naive Bayes for modeling categorical data as discrete counts, and Bernoulli Naive Bayes to handle binary-encoded features. Note that all 13 features present in the dataset were utilized in the training of each respective Naive Bayes classifier.

To train these models, we initially preprocessed the data by encoding categorical features while leaving continuous features as-is. The dataset was then split into 80% training and 20% test sets to evaluate generalization. For the Gaussian model, we tuned the var_smoothing parameter - which controls the variance added to the distribution for stability - across a logarithmic scale from 1e-9 to 1. For the Multinomial model, we tuned the alpha parameter across a range of 0.001 to 10 to handle sparse features. Finally, for the Bernoulli model, we tuned the alpha parameter across a range of 0.001 to 10 to handle binary features. From these

procedures, we documented the training and test error, as well as test accuracy for each model; we found that the Gaussian Naive Bayes model achieved the highest accuracy of 80.378%, as shown by the table below.

	Training Error	Test Error	Test Accuracy
Gaussian Naive Bayes	20.650%	19.622%	80.378%
Multinomial Naive Bayes	21.936%	20.989%	79.011%
Bernoulli Naive Bayes	26.601%	27.484%	72.517%

Decision Tree - Gary Zeng

Then we utilized a decision tree classifier by choosing a threshold on a feature and splitting the data to classify the data point into different classes. Since decision trees require the features to be numerical, we had to transform the categorical features. This includes job-type, marital status, job, relationships, race, and sex. Since we already have education as edu_num in numerical form, we decided to exclude edu.

When fine-tuning the graph, there are 3 parameters we changed: depth, min samples leaf, and min samples split. We tested a range of values between 1 to 18 for depth, min_samples_leaf, and min_samples_splits. We found that max depth of 7, a min_samples leaf of 1, and minimum split of 2 provided the most optimal balance between the training data error and testing data, as shown in the table below.

	Training Error	Test Error
Decision Tree	14.167%	14.617%

Random Forest - Gary Zeng

Additionally, we tested a random forest method, which is an ensemble method for decision trees. It splits the data into bags of data, trains one model to one bag, then adds the ensemble together. The parameters we model for this are depth, estimator, and max_features. Depth controls how many layers a tree has, the estimator parameters control the amount of decision trees in the ensemble, and max_features controls the maximum number of features used to split a node.

After tuning our model, we found that when max_depth was 6, n_estimators was 46, and max_features was 8, this produced a good balance between the training and testing errors, as shown in the table below. This is a small gap, indicating that the model is not overfitting the data.

	Training Error	**Test Error**
Random Forest	10.465%	14.156%

Results

Logistic Regression

Our Logistic Regression model had one of the lowest error rates rivaling that of our Random Forest classifier, with a resulting 14.89% error for our training folds and 14.62% on our testing data. Even among the variation of regularized C values, all error rates remained relatively the same, with variance in percentages among the training and testing rates being less than 0.03%. It seems that the change in regularization intensity did not have a substantial effect on the dataset, possibly caused by the large number of features parameterized into the model, even though each feature had varying significance on the predictive outcome. In an analysis of the most influential features, it seems that marital status, native county, and capital gain were the most substantial features that influenced the predictive outcome, having the heaviest individual

weights among the features. Though there was high bias in the data, this consensus provided a deep insight into the influential factors for income.

Naive Bayes

In terms of the performance, we found that not only the Gaussian Naive Bayes model's training error of 20.650% and test error of 19.622% were the lowest compared to the Multinomial and Bernoulli models, but it also had the highest test accuracy of 80.378%. This suggests that the dataset's continuous features were highly informative in addressing the task. Following closely is the Multinomial Naive Bayes model, with a training error of 21.936%, test error of 20.989%, and test accuracy of 79.011%. This indicates that the dataset's categorical features also held significant influence over our predictions. Lastly, the Bernoulli Naive Bayes model underperformed with a training error of 26.601%, test error of 27.484%, and test accuracy of 72.517%. This is expected, because the dataset contains mixed continuous and categorical data rather than binary features, greatly reducing its effectiveness.

Due to the outstanding performance of our Gaussian Naive Bayes model, we compared its results against those of our logistic regression, decision tree, and random forest models. We found that our Gaussian Naive Bayes model underperformed with a relatively higher training and error rate constituting a nearly 5% difference compared to the training and error rates of our other classifiers, which both sit at around an average of 14% for each.

Decision Tree & Random Forest

We found that training and testing error for random forests was less than decision trees, where random forests had a test error rate of 14.156% compared to decision tree's 14.617%. The

random forests have additional parameters that decision trees do not have, such as the number of trees. These have improved the model performance in comparison to just a single decision tree. Due to bagging, random forests had one of the lowest test error rates of not only decision trees, but also all of our other models. This follows what we learned in class lectures as the random forest is a bagging method of decision, which would produce a robust and generalized model. This approach reduces variances and mitigates the risks of overfitting the data.

Classifier	Training Error	Test Error
Logistic Regression	14.89%	14.62%
Gaussian Naive Bayes	20.650%	19.622%
Multinomial Naive Bayes	21.936%	20.989%
Bernoulli Naive Bayes	26.601%	27.484%
Random Forest	10.465%	14.156%
Decision Tree	14.166%	14.617%

Citations

Adult dataset: https://archive.ics.uci.edu/dataset/2/adult

GitHub Repository: https://github.com/gazeng2004/compsci178project