An Analysis of the 1994 United States Census

Brent Caldwell, Olivia Ryan, Teddy Amare

Introduction and Description

The following report contains an analysis of the 1994 United States Census. This dataset was originally extracted from the US Census website (<http://www.census.gov/ftp/pub/DES/www/welcome.html>), but was downloaded by us from (<https://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29>). The tasks we performed for this section are data cleaning, classification, and clustering. In the original dataset, each person is represented by their age, work-class, estimated representative weight, education, a numeric education value, marital status, occupation, relationship, race, sex, capital gains, capital losses, hours worked per week, native country, and an estimated yearly income total classified by over or under 50,000 USD over the previous year.

Data Cleaning

Before discussing the cleaning procedures performed in the code base, we would like to note that the original .data file was modified for ease of use before being processed. The names of each header were added at the beginning of the file and the separators were converted from ‘ ,’ to ‘,’ (removed spaces). The file is saved as ‘adult.csv’ and can be found contained in the submitted zip file along with this report.

When examining the data for classification and clustering tasks, we deemed that the ‘fnlwgt’ column that was used for explaining the estimated population each set of characteristics represented did not provide better context for how to classify the data set, so it was removed for ease of processing. This decision may be overturned in future iterations of the project, but for now, we agreed it did not assist in context.

There were also three characteristics that contained missing values: work-class, occupation, and native country. Work class and occupation are similar columns, in that they both are based on the types of jobs each member of the population had over the 1994 year. An analysis of the data showed that the context of the missing values in these two characteristics was that those people were unemployed, so their data was omitted. To remedy the missing values, their description was changed to ‘unemployed’. The ‘native country’ unfortunately was more difficult to pinpoint an exact reason for the missing data, so it was settled to assume the native country as the most popular selection: ‘United States’.

Classification

The classification questions of interest in this study are to predict whether a given person’s income is greater than or less than/equal to $50,000 based on 12 different variables. These dependent variables include age, work class, education, marital status, occupation, relationship, race, sex, capital gain, capital loss, hours per week, and native country. This is a binary classification problem with two class labels: <=50k and >50k.

The classification task was preformed using decision trees using all the appropriate features discussed above. The model’s performance was evaluated on three different test-training splits including 80-20, 60-40, and 50-50 splits. The model’s accuracy on the test set was recorded for each split, along with its accuracy on 5-fold and 10-fold cross validation. The goodness of the models learned from the dataset were evaluated using a classification report which provides more detailed information about the model’s precision, recall, and F1-score. These metrics help to determine the effectiveness of the models at correctly classifying individuals as having an income above or below $50,000.

Overall, the model preformed reasonably well on all three splits. All three splits generated an accuracy of over 0.85. In other words, the model correctly classifies over 85% of the instances in the test set. The 80-20 split preformed the best with an accuracy of 0.86. The 60-40 split and 50-50 split both generated an accuracy of 0.85 with the 60-40 split being slightly more accurate than the 50-50. We can see that the accuracy on 5-fold and 10-fold cross validations are very similar, with only a small difference between them. This suggests that the model’s performance is relatively stable across different folds of the data.

According to the classification report, the 80-20 split achieved the highest precision, recall, and F1-score for both the <=50k class and the >50k class. In all splits, the precision, recall, and F1-score for the <=50k class is higher than the >50k class, which indicates that the model is better at predicting the <=50k class. This discrepancy may be due to an imbalance in the dataset where the number of instances in the <=50k class is higher than the >50k class. Overall, the classification reports suggest good results for all the goodness of fit metrics for the <=50k class. The high precision indicates that the model has a low false positive rate; meaning that when it predicts a positive instance, it is usually correct. The high recall indicates that the model has a low false negative rate, meaning that it can identify most of the positive instances. The high F1-score further indicates great precision and recall.

(The following screenshots were taken from a version of the code that included binning the age segments. Current results may vary slightly. The accuracies remain similar)

A picture containing calendar

Description automatically generated

Diagram

Description automatically generated

Clustering

Clustering type that was performed is called K-Means. This clustering approach is an unsupervised machine learning algorithm to partition our data set into k clusters based on their similarity. To obtain the optimal k, we have implemented two algorithms: elbow method and silhouette scoer.

The elbow method is completed by plotting Within-Cluster sum of squres against the number of clusters. We find the “elbow” point by finding the dip where the graph changes it’s shape.

The silhouette method finds the optimal k by calculating the mean silhouette coefficient for different values of k. The higher the score, the better the clustering, meaning the points are matched to their assigned cluster. Before completing the clustering, we dropped the education attribute because there was another attribute that had the same numerical values instead of the categorical one.

Using both methods, we find the optimal k to be 3 which was consistent across multiple iterations.

k=2, the clusters are primarily separated by Age, where one of the clusters appears for younger individuals vs older ones. We also see overlap between Education and hours per week between the two clusters.

k=3, we separate the clusters by Age and Education and we can see more into younger population with higher level of education and the opposite representing the higher level. Some attributes don’t really have a big effect on the clustering. Relationship attribute can be dropped for example in our case.

The clustering has provided insights to the dataset that we wouldn’t normally see. Clusters have showed us different population ranging from a certain age with the relationship they have with other attributes.

Chart, scatter chart

Description automatically generated

Chart, scatter chart

Description automatically generated

Conclusion

This report contained a description and outline of the data cleaning, classification, and clustering segments for the analysis of the 1994 US census.

Lessons:

In terms of data cleaning, there also was an attempt at binning the age variable into different life stages. However, this approach didn’t provide any more accuracte results, nor did it work while trying to cluster data in the clustering segment. As much as it was useful duing classification, the bins were eventually commented out.

In terms of what the data gives us, it seems fairly easy to pinpoint what a particular person’s net income is by virtue of their job and marital status, as evidenced by how precise and accurate the different training and test splits ended up during the classification tests.

Questions:  
An economic term we hear a bit about is the top ‘1%’, as in the people with the most net worth. We wonder what their representation looks like in the dataset.