US Census Data

Daniel Friar

July 19, 2017

1 Exploratory Analysis and Data Cleaning

We first load the train and test datasets, assign column names from the metadata file and remove duplicate rows, leaving 196,000 training observations and 99,000 test observations. We then view the distribution of classes in the training set which indicates it is heavily imbalanced in favour of the 'Less than 50,000' class.

Class	Number of Observations
Less than 50,000	183,912
50,000+	12,382

Table 1.1: Class Distribution

We check for *NA* values and find there are none present in the dataset, however we note that several of the categorical variables contain a '?' class corresponding to missing data. In particular, it's noted that the three 'Migration code' variables are missing values in approximately 50% of cases.

1.1 Numeric Variables

We then look at the distribution of the numeric variables in the dataset, shown in figure 3.1 below.

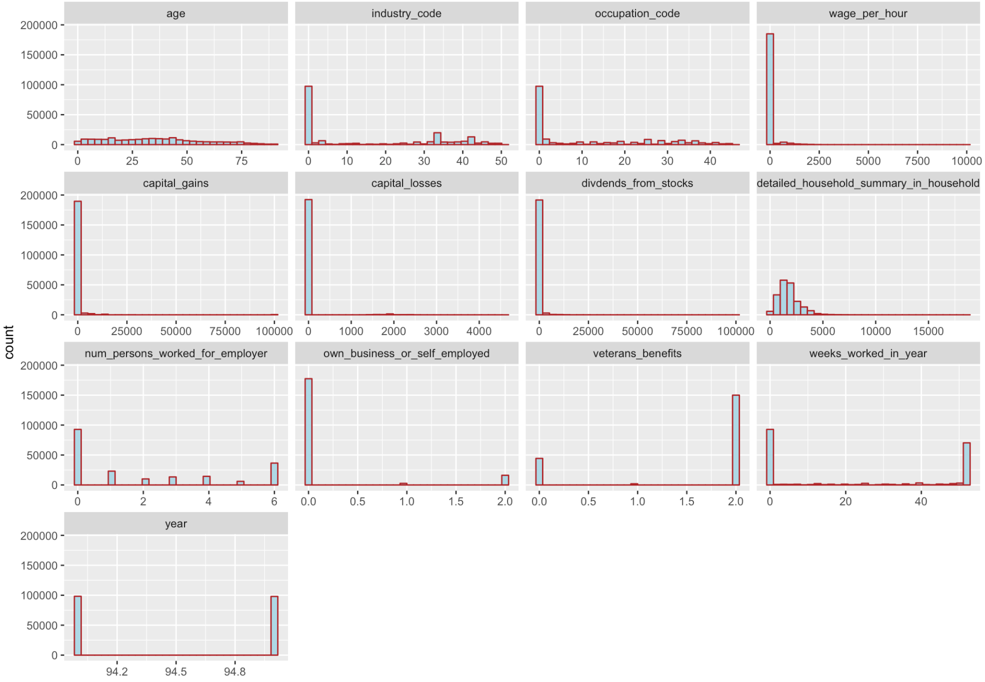

Figure 1.1: Distribution of numeric variables.

The figure indicates that ‘year’, ‘veterans benefits’, ‘num persons worked for employer’ and ‘own business or self employed’ may be better off as factor variables so these are converted. It’s also noted that ‘industry code’ and ‘occupation code’ may be better off as factor variables, but we leave these as numeric in case there is some information in the numerical ordering i.e. similar industries may have codes with similar numerical value. Since categorical variables will need a one-hot encoding, this also reduces the number of features fed into the model.

‘Capital gains’, ‘capital losses’, ‘dividends from stocks’ and ‘wage per hour’ all have long tails so we visualize these distributions with values of 0 removed in order to spot outliers.

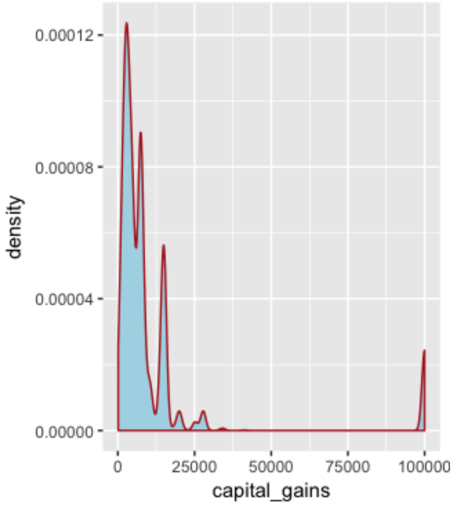

Figure 1.2: Capital Gains Distribution.

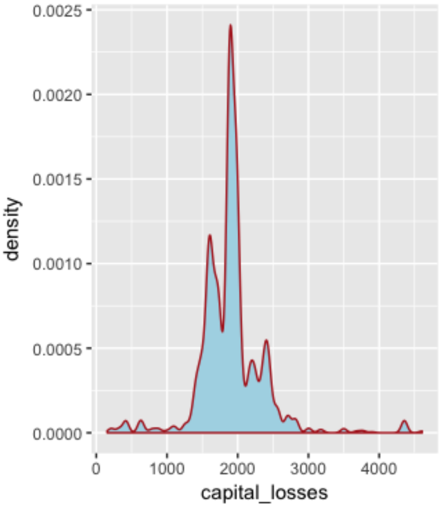

Figure 1.3: Capital Losses Distribution.

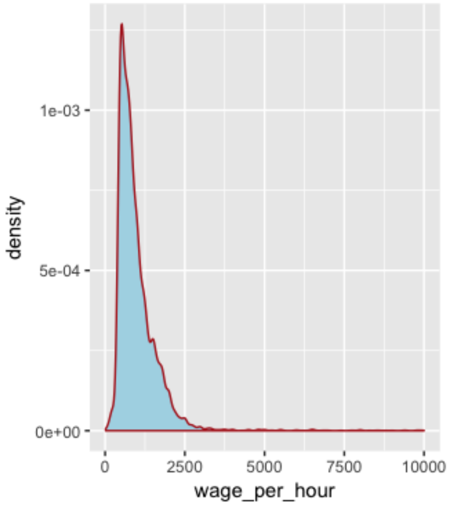

Figure 1.4: Wage per Hour Distribution.

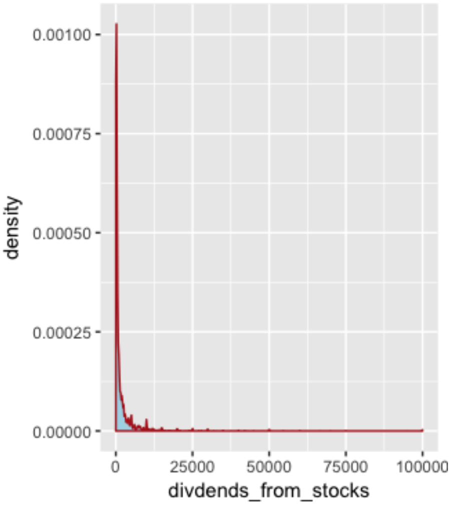

Figure 1.5: Dividends from Stock Distribution.

The ‘wage per hour’ and ’dividends from stock’ plots indicate very long tails with possible outliers. We remove observations with ‘wage per hour’ higher than 5,000 and with ‘dividends from stock’ higher than 25,000, corresponding to the removal of just over 200 observations in total.

1.2 Categorical Variables

We go through the categorical values in turn and simplify these in some cases. In particular, we reduce the number of classes for the ‘education’ variable and simplify the ‘state of previous residence’ variable to {‘Householder’, ‘Not Householder’}. For larger datasets, further aggregation of categorical variables may be useful to reduce model training time, in particular ‘country of birth’ could be split into {‘U.S’, ‘Outside of U.S.’} and the ‘region of previous residence’ variable dropped, to avoid introducing several features. The categorical variables with missing values are left in the dataset, with missing values encoded as an extra class. Before training the model, we form a one-hot encoding of the categorical variables.

2 Model Selection

2.1 Validation Set and Balancing Classes

We further split out 25% of the training data in order to create a validation set of approximately 50,000 rows.

In order to address the class imbalance in the resulting 150,000 row training set, we sample just 50% of records from the majority “Less than 50,000” class and include the “50,000+” examples four times, to give a more balanced training set split as shown below.

Class	Number of Observations
Less than 50,000	68,956
50,000+	45,660

Table 2.1: Class Distribution

Since the class distribution in the original data is heavily skewed towards the ‘Less than 50,000’ class, an evaluation metric other than prediction accuracy should be used (since predicting the majority class every time would give roughly 90% accuracy but is clearly not a meaningful classifier). To account for this we instead judge performance using the log loss, which takes into account the confidence in each predicted label, along with the area under the ROC curve.

2.2 Random Forest Model and Grid Search

We fit a random forest model to the data and run a grid search over several parameters in order to choose the best configuration. A cross-entropy evaluation metric is used, with performance judged by 3-fold cross-validation on the training set. The chosen parameters are a minimum of 3 samples per leaf, a max depth of 20 and max of 30 features at each split. We also run for 500 trees.

After choosing parameters, we measure performance on the validation set. While a validation set may not appear to be necessary in this case, it’s useful for judging performance with future models i.e. if we were to fit several different classifiers, validation performance may be used to choose between them. Results are as follows:

$$\begin{aligned} Logloss &= 0.216 \\ AUC &= 0.945 \\ Accuracy &= 0.906 \end{aligned} \tag{1}$$

3 Evaluating Performance on Test Set

After including the 3,000 positive (50,000+) examples from the validation set and re-training, test set performance is as follows:

$$\begin{aligned} Logloss &= 0.212 \\ AUC &= 0.948 \\ Accuracy &= 0.908 \end{aligned} \tag{2}$$

The minority class is predicted in around 12% of cases, indicating that the re-sampling prevented the model from trivially always predicting the majority class.

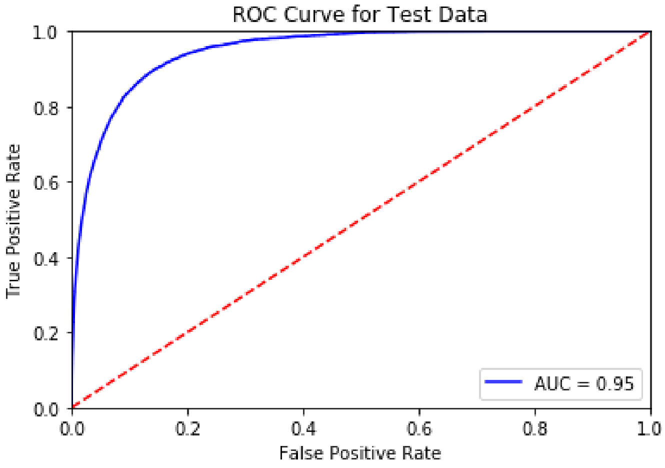

Figure 3.1: ROC curve for predictions on test data.

4 Feature Importance

In order to judge feature importance we look at the mean decrease in the performance metric by feature (over the 500 trees), which produces the plot in figure 4.1.

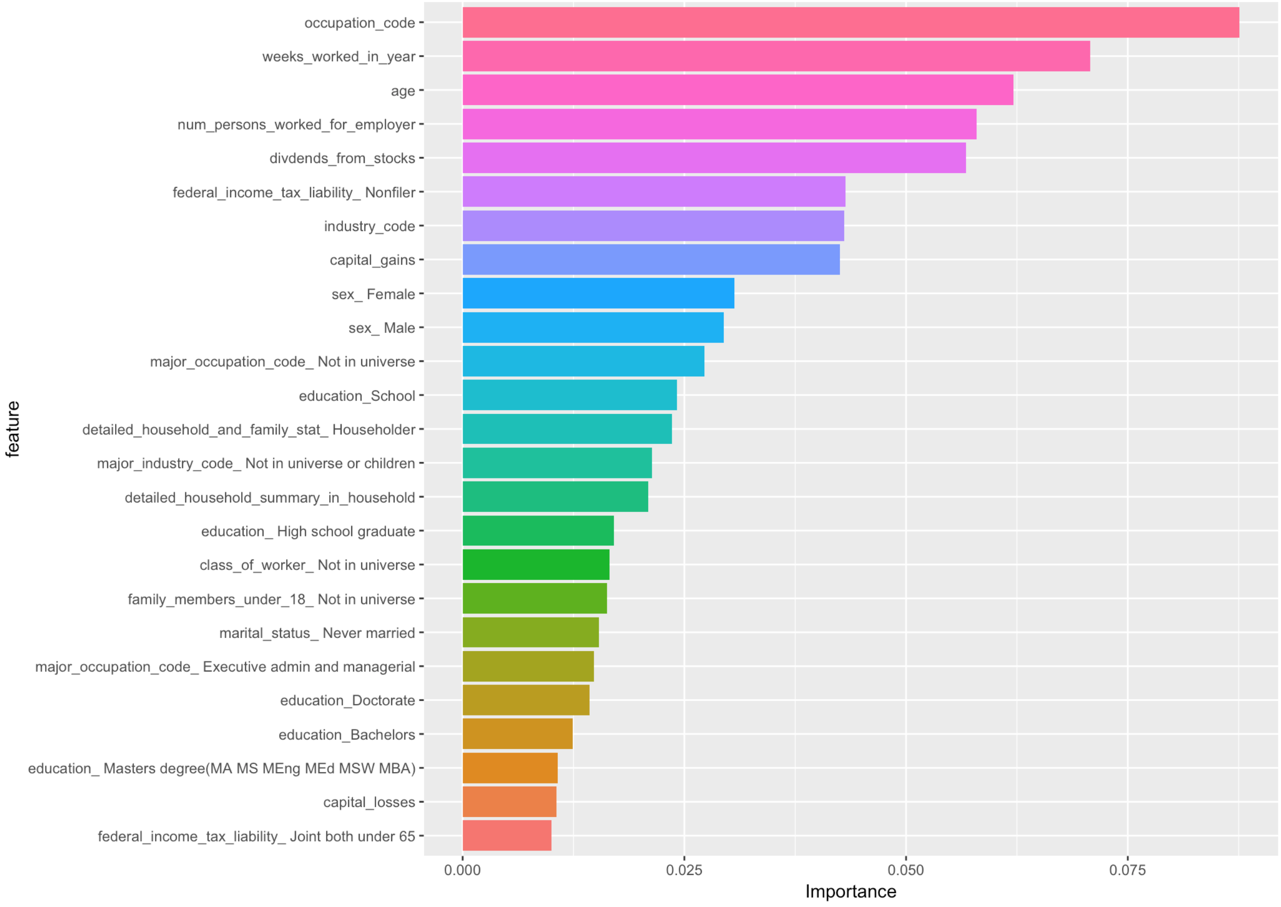

Figure 4.1: Feature importance for 25 most importance features.

As may be expected, the figure shows that ‘occupation code’, ‘age’ and ‘weeks worked in a year’ all have a significant impact in the model. The ‘num person worked for employer’ variable is also judged to be important, which is possibly the case as it also encodes information as to whether or not the worker is self-employed (i.e. this variable is greater than 0 if the worker is self-employed). As expected, ‘sex’ and ‘education’ also play significant roles.

We view some of these variables in more detail below.

4.1 Occupation Code

We compare the percentage of observations with salary above 50,000 for each of the observation codes, with figure 4.2 showing the 20 occupation codes with the highest proportions.

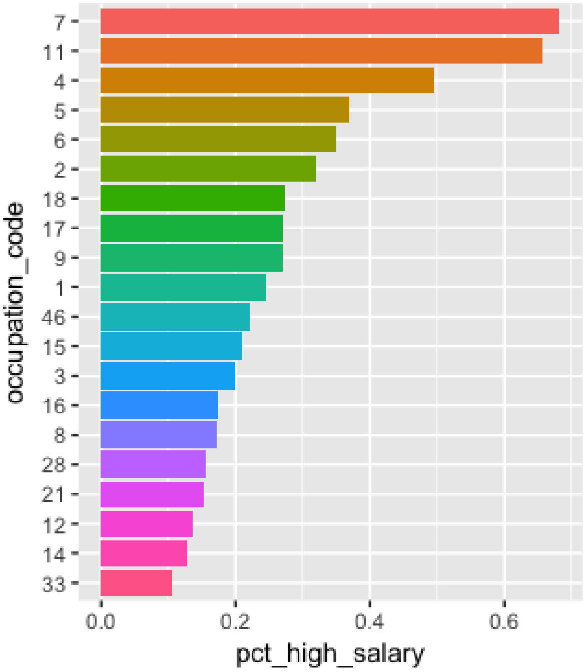

Figure 4.2: Proportion with high salary for top 20 occupation codes.

4.2 Weeks Worked in Year

The table below shows the difference in weeks worked in the year across the two classes, indicating a large difference, with those earning more than 50,000 working significantly more as expected.

	50,000+	Less than 50,000
Min	0	0
1st Q	52	0
Median	52	0
Mean	48.3	21.9
3rd Q	52	52
Max	52	52

Table 4.1: Weeks worked in year distribution across classes.

4.3 Age

The boxplots in figure 4.3 below show the differences between the two classes, indicating that older people are more likely to earn above 50,000.

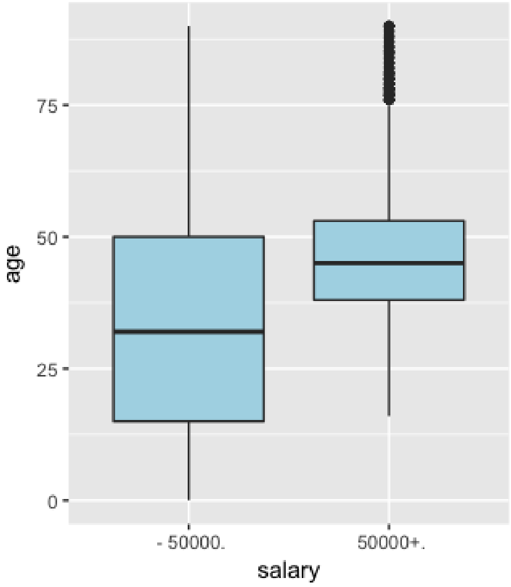

Figure 4.3: Boxplots for age across the two classes.

5 Difficulties and Future Extensions

The data is reasonably clean in this case and the main difficulty is in dealing with the large class imbalance. While naively training a classifier may lead to good accuracy, it is likely to always predict the majority class which makes the sampling procedure described above necessary. As an extension, more sophisticated ways of dealing with this imbalance could be used, such as a custom loss function which weights the predictions according to the class weighting or synthetically including examples from the minority class. These approaches would avoid sampling from the training data.

A random forest model was used because of it's interpretability and ease of training. It's also good at dealing with categorical variables, which are prominent in this dataset. Better performance may be obtained with a different classifier, in particular a Gaussian Kernel SVM may achieve good results as it's able to capture interactions between the variables. Alternatively, further feature engineering may be performed in order to include the effect of variable interactions.

An alternative extension may be to just remove variables from the model that are not observed to be important, in order to improve interpretability and possibly model performance.

For applying the model at scale, it may be useful to further group some of the categorical variables in order to reduce the number of features in the model to make training faster.