US Census analysis

Reda Affane

December 30, 2016

1 Descriptive analysis

1.1 Missing values

In this section we will clean the dataset by looking at the missing values for each column and will try to pick the right method to fill in the wholes.
The section below shows the percentage of missing values for each attribute:

Variables sorted by number of missings:

Variable	Count
migration code-change in msa	0.499671717
migration code-change in reg	0.499671717
migration code-move within reg	0.499671717
migration prev res in sunbelt	0.499671717
country of birth father	0.033645244
country of birth mother	0.030668144
country of birth self	0.017005558
state of previous residence	0.003548463
age	0.000000000
class of worker	0.000000000
detailed industry recode	0.000000000
detailed occupation recode	0.000000000
education	0.000000000
wage per hour	0.000000000
enroll in edu inst last wk	0.000000000
marital stat	0.000000000
major industry code	0.000000000
major occupation code	0.000000000
race	0.000000000
hispanic origin	0.000000000
sex	0.000000000
member of a labor union	0.000000000
reason for unemployment	0.000000000
full or part time employment stat	0.000000000
capital gains	0.000000000
capital losses	0.000000000
dividends from stocks	0.000000000
tax filer stat	0.000000000
region of previous residence	0.000000000
detailed household and family stat	0.000000000
detailed household summary in household	0.000000000
instance weigth	0.000000000
live in this house 1 year ago	0.000000000
num persons worked for employer	0.000000000
family members under 18	0.000000000
citizenship	0.000000000
own business or self employed	0.000000000

```
fill inc questionnaire for veteran's admin  0.000000000
                        veterans benefits   0.000000000
                    weeks worked in year    0.000000000
                                    year    0.000000000
                                  income    0.000000000
```

Most of the attributes don't contain any missing value, except for 8 attributes, 3 of which have almost 50% of missing values.
For the attributes that contain 50% of missing values (migration code-change in msa, migration code-change in reg, migration code-move within reg, migration prev res in sunbelt), we choose to drop them (the whole column).
As for the other ones (country of birth father, country of birth mother, country of birth self, state of previous residence), we choose to replace the missing values in each attribute by the most frequent value. We chose this approach for its simplicity. It comes of course with a drawback, which is reducing the variability within the attributes. This reduction in variability will not be very harmful for our model since the proportion of missing values don't exceed 3.5% at most.

1.2 Formatting

Some values are coded in the dataset as continuous values, while clearly being categorical. We are going to convert these variables to the correct format so that we ultimately have 33 categorical variables and 7 continuous variables as stated in the metadata file.

1.3 Descriptive analysis

Let's have a look at the target variable:

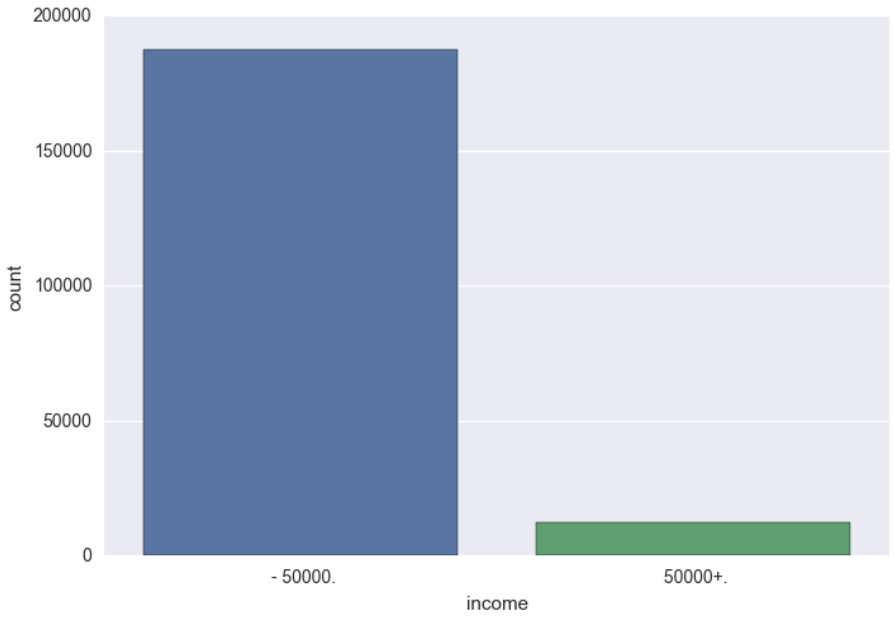

Figure 1: income distribution
+50000: 6.2%
-50000: 93.79%

The +50000 income makes only 6.2% of the total inputs, while the -50000 income make 93.79%. This is a case of an unbalanced dataset. This means that if we classify all the inputs as "-50000", we can get an accuracy of 93%. This shows that the accuracy measure isn't quite adapted to this kind of heavily skewed dataset. We shall instead use the Roc curve and the Area under the ROC curve to asses our models.

1.3.1 Numerical features

The dataset contains 7 numerical features, let's explore some of them.

- **Age**:
  The following figures show the distribution of age within the population:

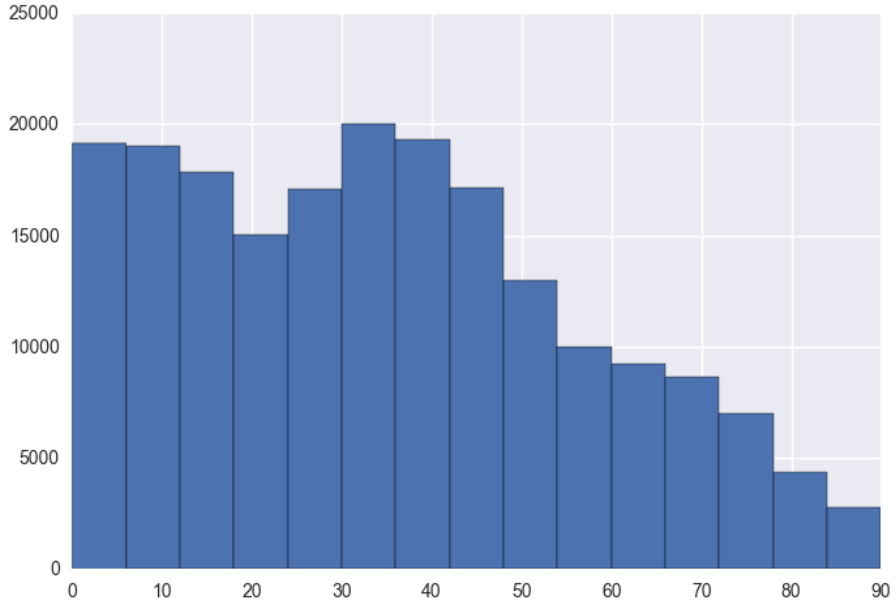

Figure 2: Distribution of the age

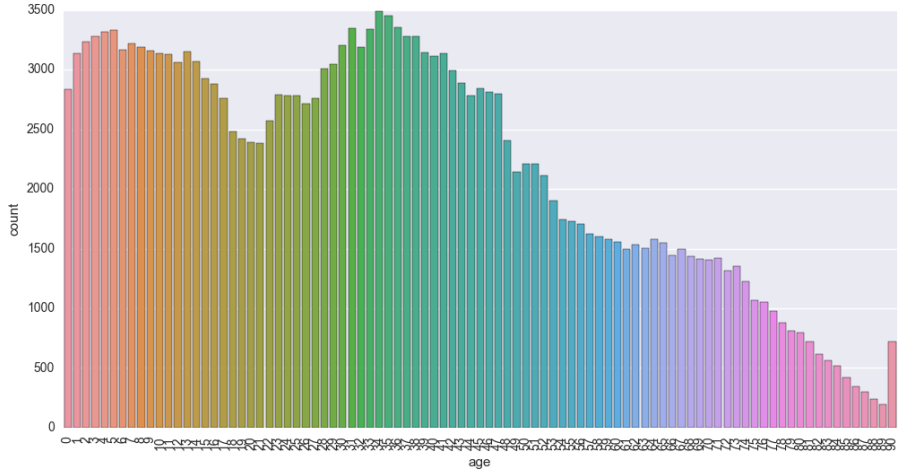

Figure 3: Count plot of age values

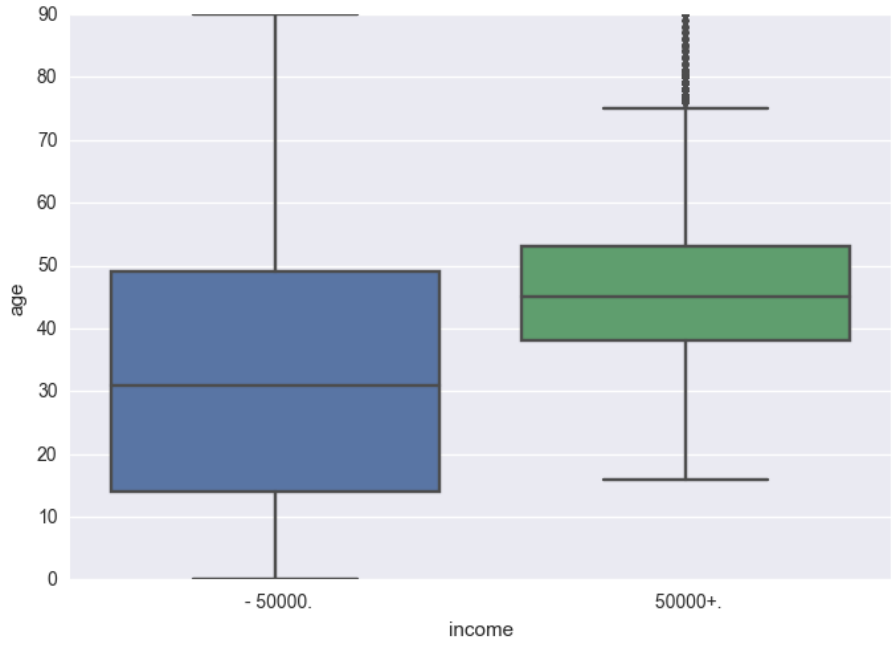

Figure 4: Boxplot for the variable age

We can see that there are 2 pics in the population age, one around 35 yo and the other one around 6 yo. Figure 3 suggests that the people that earn more that 50.000 are older in average than the people who don't.
We can also see that this data set include all the individuals, including the children that obviously have no income. We can therefore create age ranges to help whichever classifier we'll use later in the classification. We choose to consider 3 classes: age below 20 (coded as 0), age between 21 and 61 (coded as 1), and age above 62 (which is the minimum retirement age in the united states) (coded as 2).
The following figure shows how this classification of the age variable is correlated with the income.

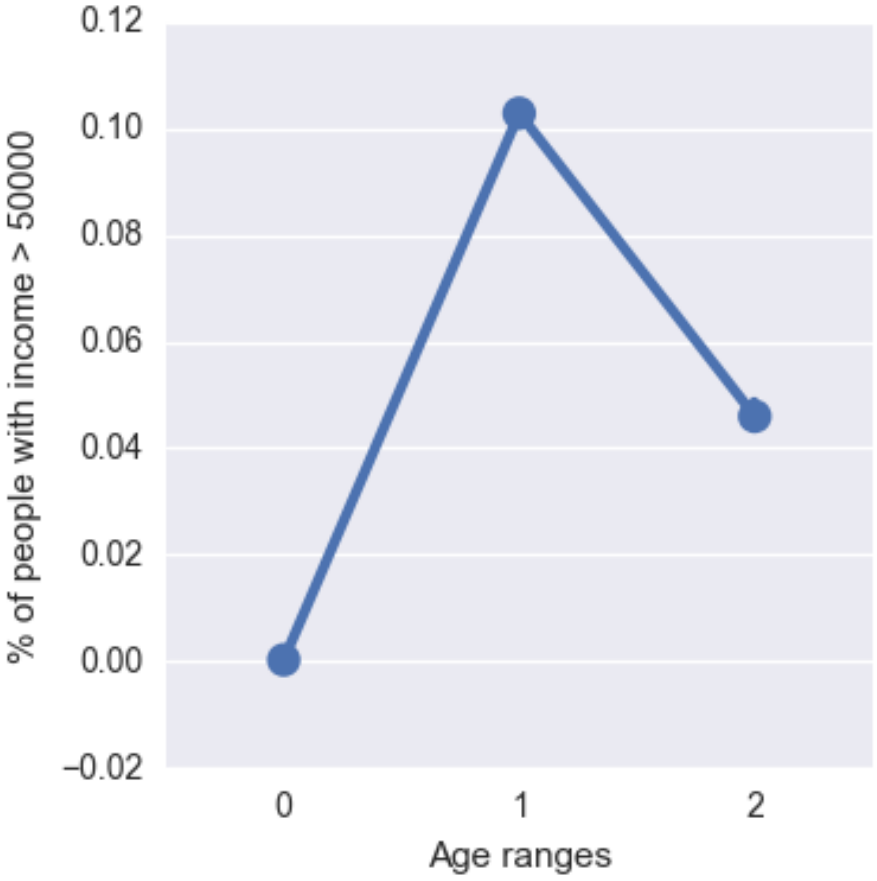

Figure 5: proportion of people earning more that 50000 across age ranges

- **num persons worked for employer**:

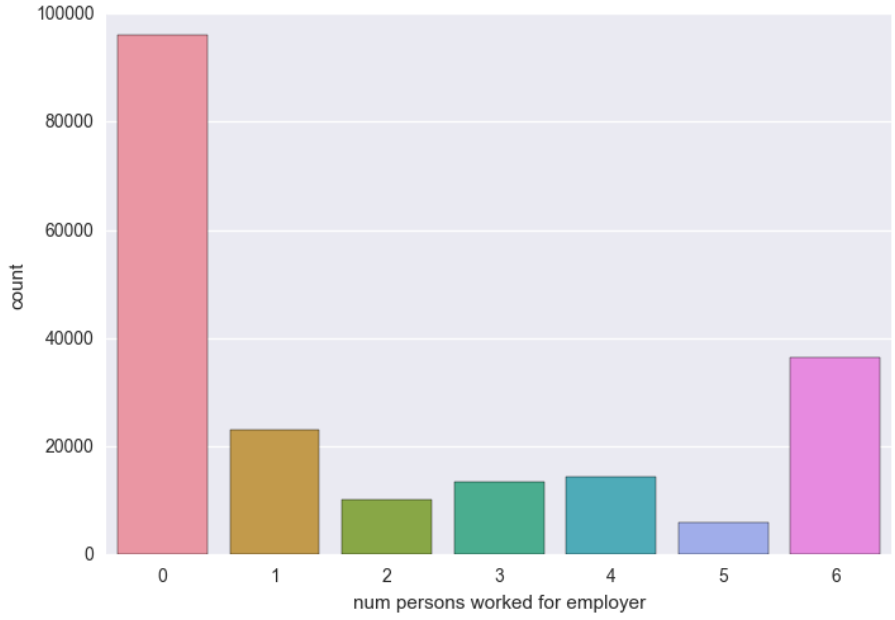

Figure 6: Count plot of number of workers

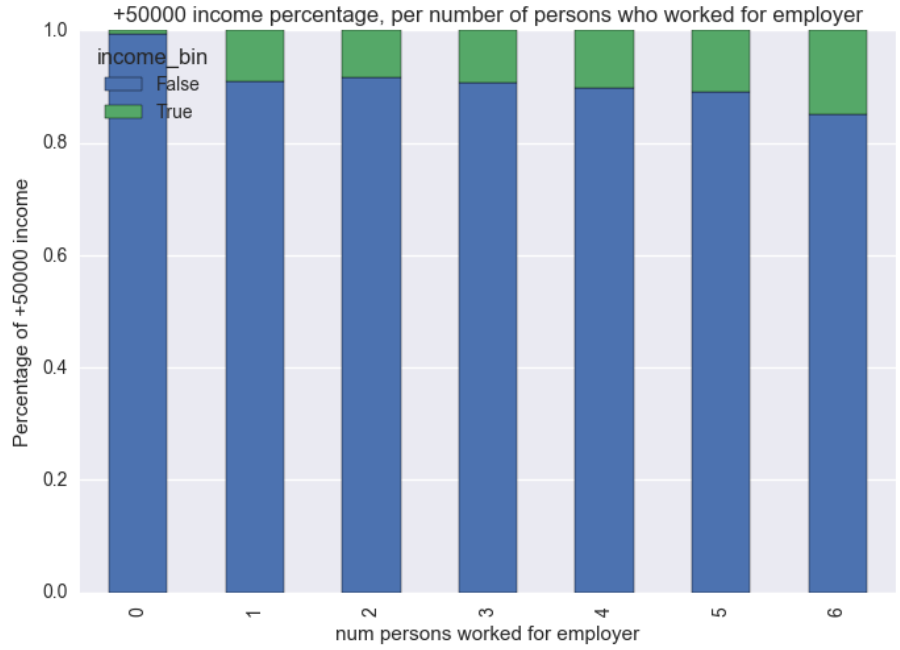

Figure 7: Relation between the number of workers and the income

The meaning of this variable is not quite clear. It could represent the number of persons working for the person that we are considering, or the number of employees on that person's team, or something else. Either way, figure 6 suggests that this variable is correlated with the income.

1.3.2 Categorical features

This data set contains 33 categorical variables, let's look at some of them:

- **Race**:

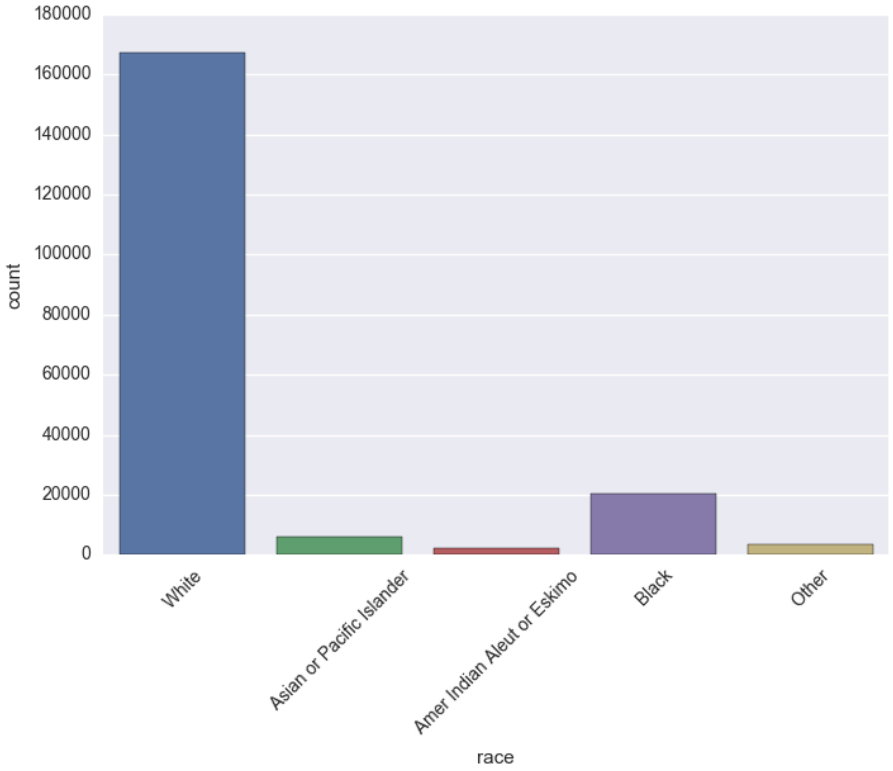

Figure 8: Distribution of race

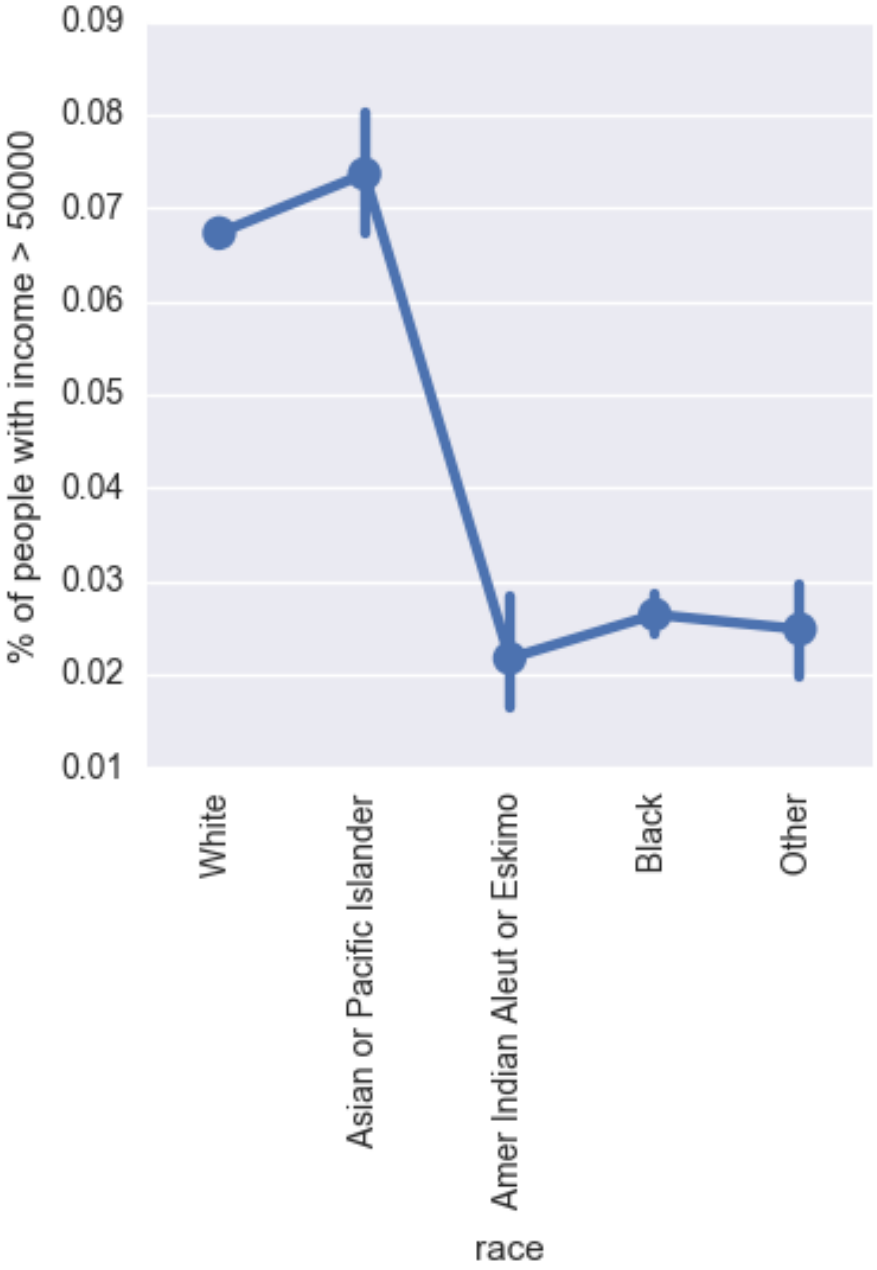

Figure 9: Relation between the race and the income

Most of the persons in the data set are white people, followed by black people. From figure 8, we can see that the race may play a role in the classification, since "white" and "Asian or Pacific islanders" seem to have higher income.

- **Sexe**:

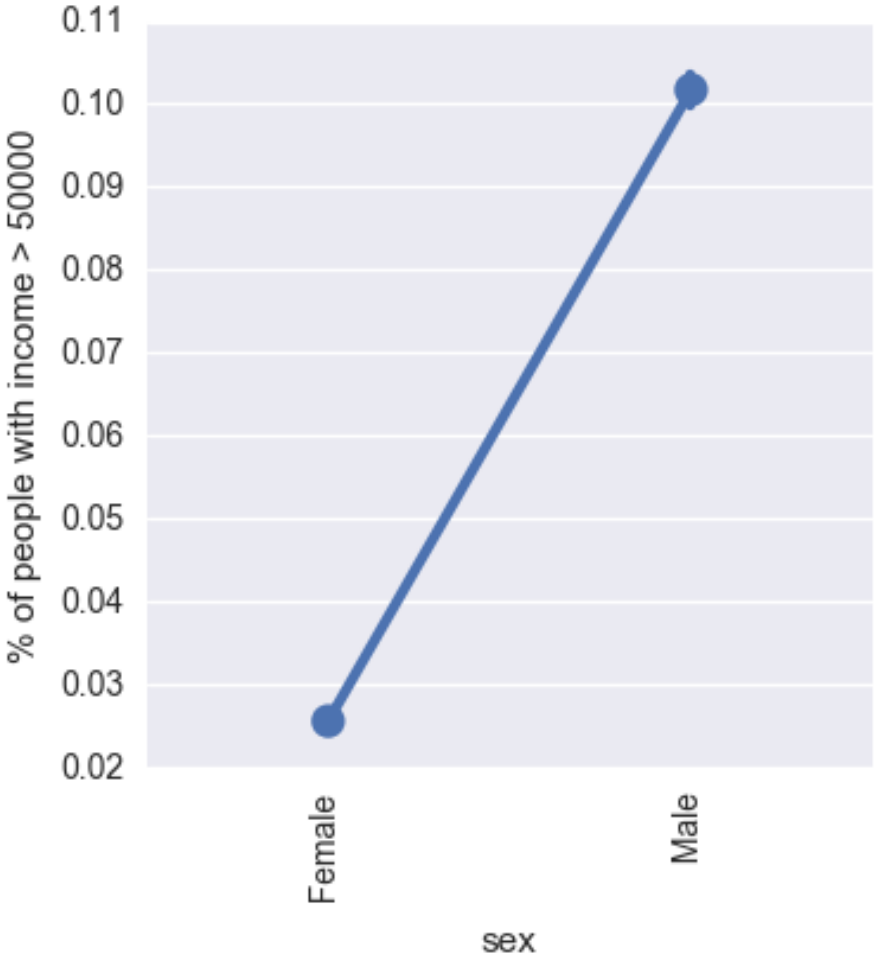

Figure 10: Relation between sex and income

- **Member of a labor union**:

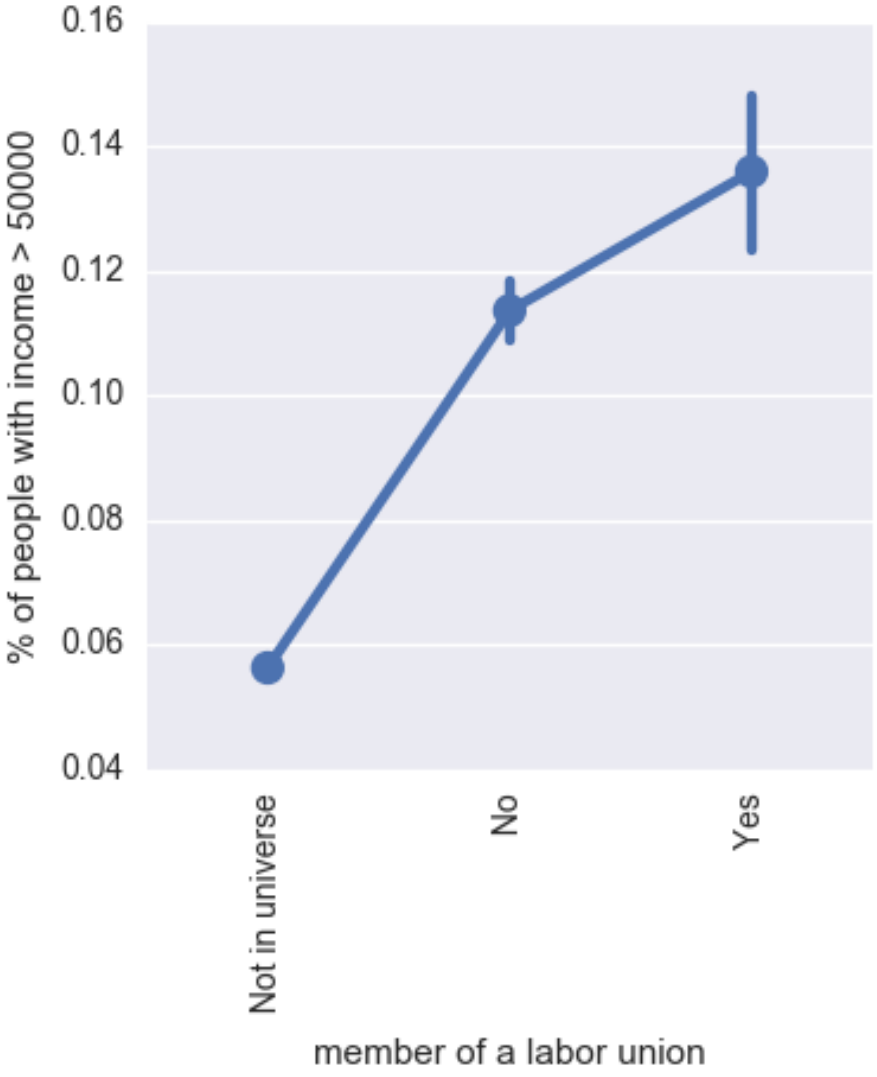

Figure 11: Relation between membership in a labor union and the income

2 Classification

In this section, we're going to try to achieve 2 goals:

- The first one is to explain the model and select the most statistically significant features for the model and see how they correlate to the income.
- The second goal is to try different models and compare their performance.

2.1 Explain the model

We choose to apply a logistic regression model to our dataset with the glm function in R. Here are the results:

- How well did we explain the model ?

```
Null deviance: 69603  on 149641  degrees of freedom
Residual deviance: 36559  on 149244  degrees of freedom
AIC: 37355

Number of Fisher Scoring iterations: 23
```

Which gives us an R^2 of 34.4%

Our model contains 41 variables, of which 33 are categorical, each one of them having several levels. This makes a total of 383 features that are used by the logistic regression model, after converting categorical variables into dummy variables. As the result of the regression shows, most of these variables are not significant to the model. In the rest of this section, we will try to reduce the number of variable in order to make the model more interpretable.

2.1.1 Feature Selection

There are plenty of ways to perform feature selection. We will apply the Lasso, as a feature selection method and a regularizer. We were able to select 157 variables from the total.

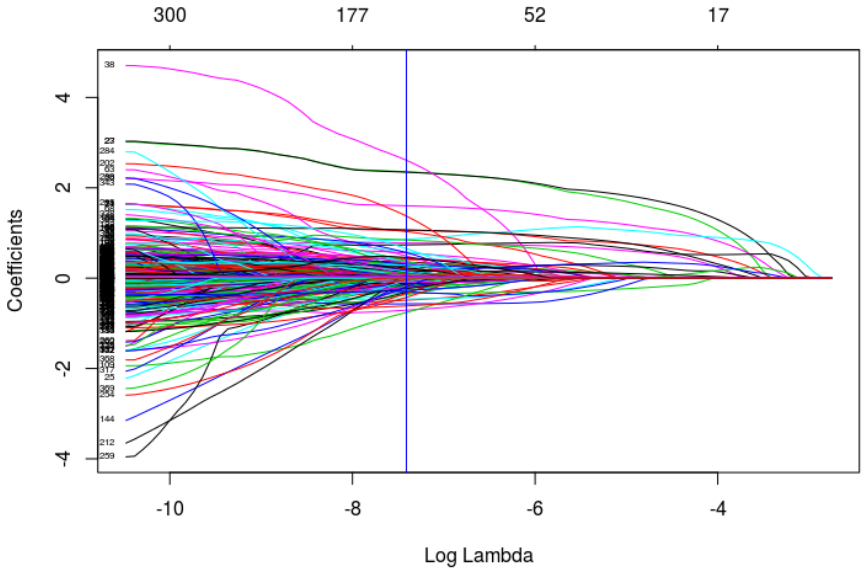

Figure 12: Evolution of the coefficients with lambda

The figure above shows how the coefficients shrinks with the value of λ, the regularization coefficient. The vertical line shows the value of λ that we selected. This value was chosen by using cross-validation over a range of lambdas.

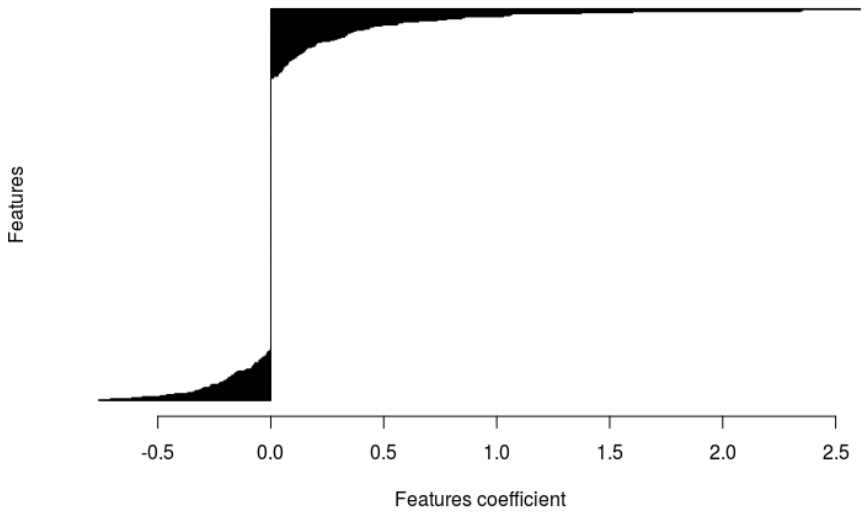

Figure 13: The value of the coefficients

This figure is a visualization of all the coefficients (383) of the model. We can see that most of them are set to 0. The features that the model picked are presented in the next paragraph.

2.1.2 Interpretation

Several features seem to be highly correlated with the level of income:

- class of worker: "self employed incorporated" workers are more likely to earn more than 50000, unlike the "local government" and "state government" workers who are more likely to earn less.
- Marital Stat: "Never married" are more likely to earn less than 50000
- Member of labour union: members of labour unions are more likely to earn more than 50000 than those who are not.
- Sex: Male workers are more likely to earn more than 50000 than female workers.
- Capital gains, capital losses and dividends from stock are also highly correlated with the income. The higher these attributes are for a person, the more likely he will earn more than 50000.
- tax filer stat
- num persons worked for employer
- own business or self employed
- full or part time employment stat
- age

2.2 Prediction accuracy

We applied 4 different models to the dataset: a logistic regression model, a logistic regression model with tuned weights (this is a built-in function in the scikit-learn model, used to deal with skewed datasets), a random forest classifier and adaboost. We split the training set into a training set and a validation set, using the train_test_split function of Scikit-kit learn with stratification.

Below are the results:

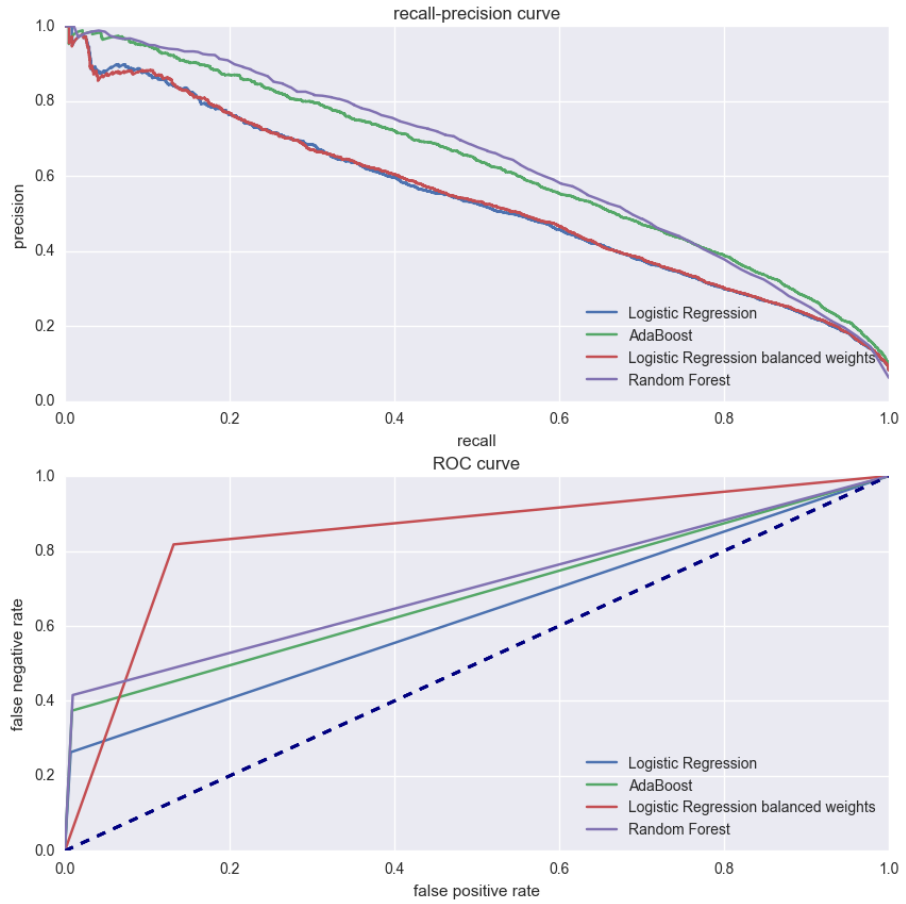

Figure 14: Recall-precision curve and ROC curve

```
Accuracy:
{'Logistic Regression': 0.94754187992649186, 'AdaBoost': 0.95293349330984312,
'Random Forest': 0.95469525993651561,
'Logistic Regression balanced weights': 0.86481478668954936}
```

```
AUC:
{'Logistic Regression': 0.92750956409443031, 'AdaBoost': 0.94565805540514292,
'Random Forest': 0.93782984929541358,
'Logistic Regression balanced weights': 0.92851866255749782}
```

The best model is the adaboost model with an AUC of 0.94 and an accuracy of 0.95.

There are several methods that try to tackle the problem of skewed datasets. The most common ones are oversampling the minority class, undersampling the majority class or a combination of the two. We try two methods: random oversampling and SMOT.

- Random oversampling:

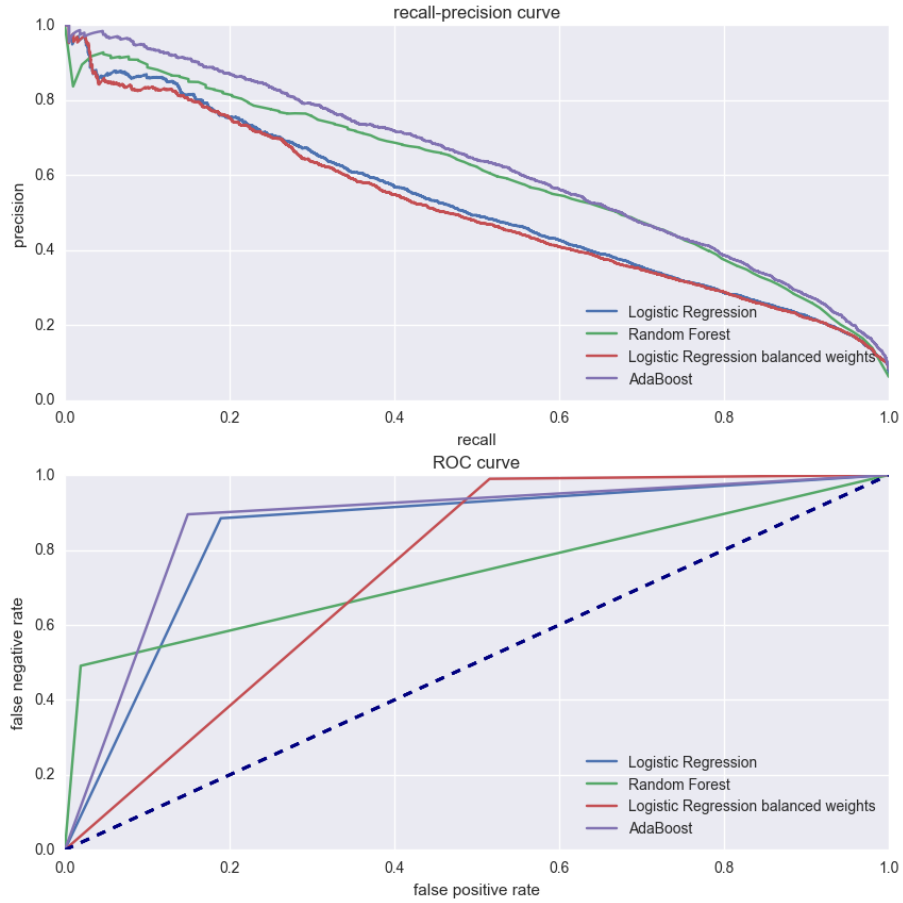

Figure 15: Recall-precision curve and ROC curve

```
Accuracy:
{'Logistic Regression': 0.81528788177938427, 'Random Forest': 0.95038196922983464
'Logistic Regression balanced weights': 0.51580274288838601,
'AdaBoost': 0.85351518004951177}
```

```
AUC:
{'Logistic Regression': 0.92361582351989191, 'Random Forest': 0.93719230244387575
'Logistic Regression balanced weights': 0.922114514380732,
'AdaBoost': 0.94607873138900211}
```

This method method doesn't seem to improve the performance by much.

- SMOT

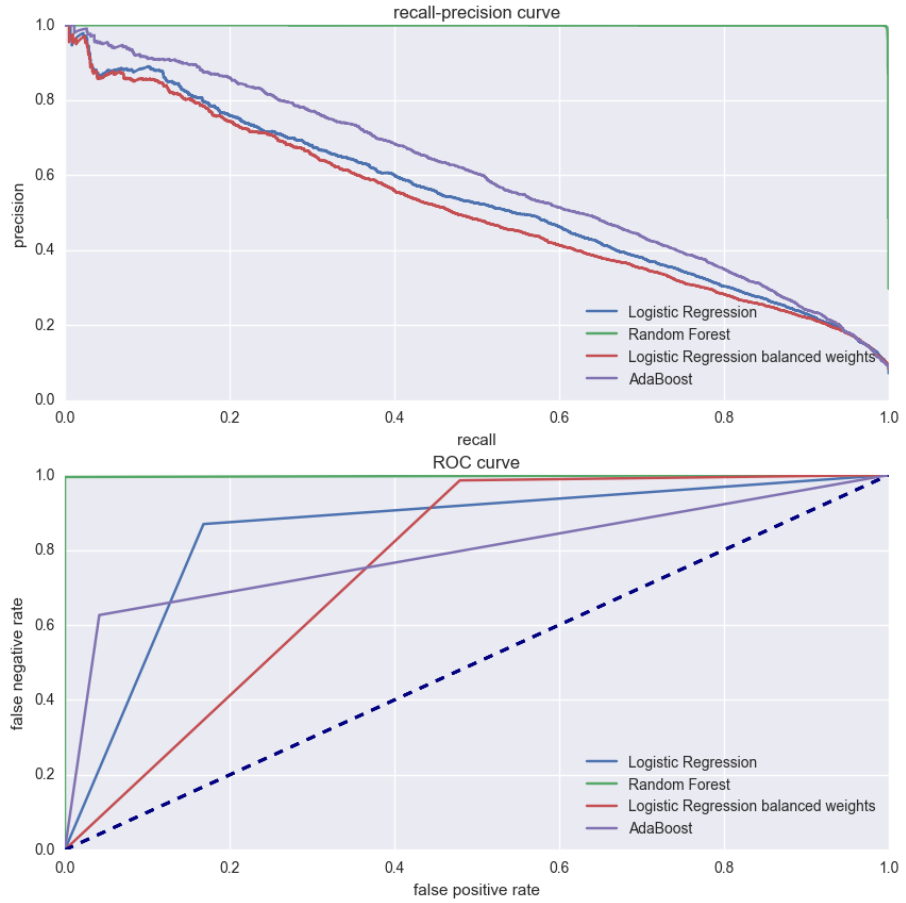

Figure 16: Recall-precision curve and ROC curve

```
Accuracy:
{'Logistic Regression': 0.83401424600944674, 'Random Forest': 0.99955955834333188
'Logistic Regression balanced weights': 0.54945856051516484,
'AdaBoost': 0.93757878589979193}

AUC:
{'Logistic Regression': 0.92794437185561518, 'Random Forest': 0.99989729891326951
'Logistic Regression balanced weights': 0.92235338738742134,
'AdaBoost': 0.93558949892751864}
```

The Random forest classifier performs weirdly well. We might have over fitted it.

2.2.1 Results

Finally, let's try the last models with the SMOT procedure on the test set (that we didn't touch so far).

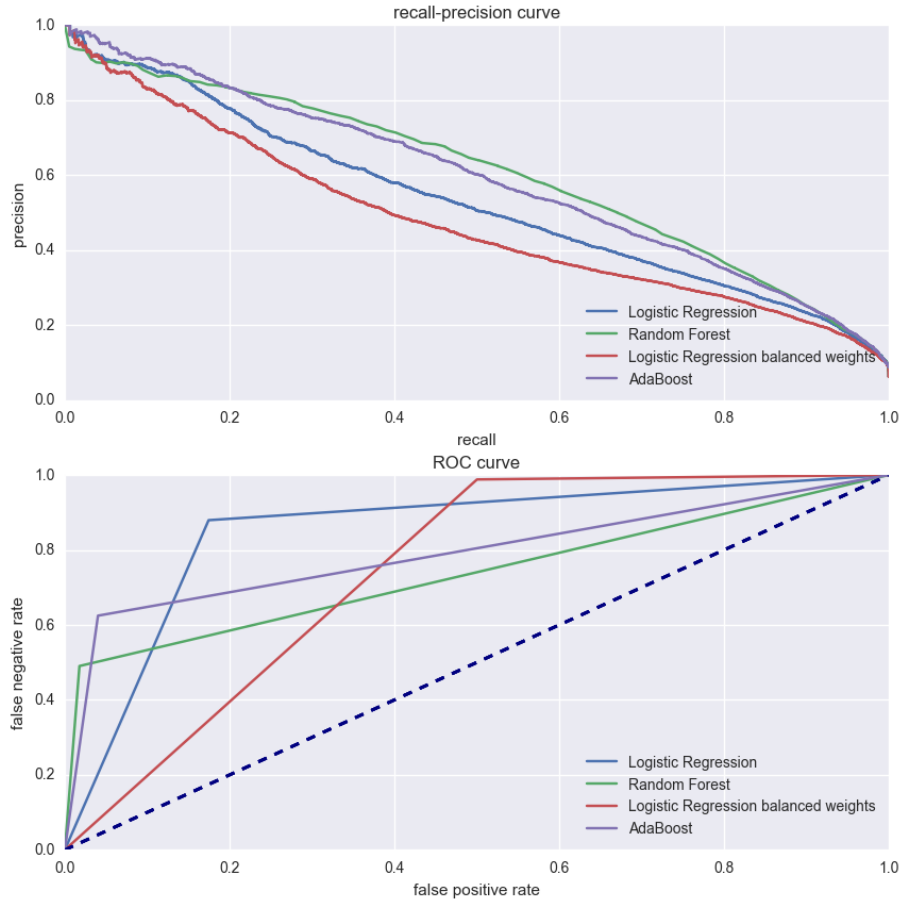

Figure 17: Recall-precision curve and ROC curve

```
Accuracy:
{'Logistic Regression': 0.82884264549628117, 'Random Forest': 0.95173512960846818,
'Logistic Regression balanced weights': 0.5300515226238447,
'AdaBoost': 0.93898478378540928}

AUC:
{'Logistic Regression': 0.92749502533540273, 'Random Forest': 0.94001410210410141,
'Logistic Regression balanced weights': 0.91518265444581515,
'AdaBoost': 0.93784553364798706}
```

The best model is the Random Forest model with an AUC of **0.94** and an accuracy of **0.9517**