The dataset is loaded with `pandas`, it contains 48842 rows:

```
df = pd.read_csv("adult-income.csv")
```

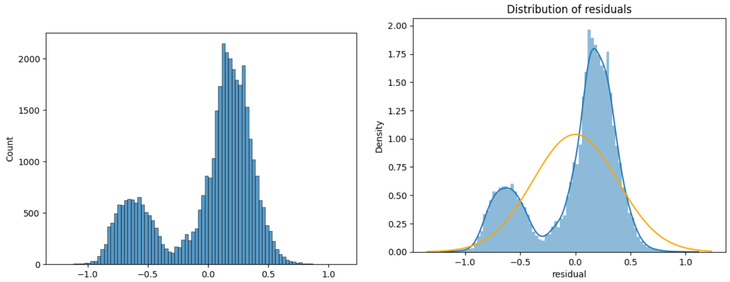
We choose the columns `educational-num`, `age` and `hours-per-week` as independent variables. This will constitute our vector *X*. Then the goal is to predict the column `income`. This column contains only two values, it is a binary column. The values are `<=50K` and `>50K`. After splitting the dataset into train and test subsets with the function `train_test_split`, we define our vectors, and we prepare for regression.

```
y_train = train['income']
X_train = train[['educational-num', 'age', 'hours-per-week']]
```

We will apply three models to the data:

1. Linear regression
2. Decision tree
3. Random forest

After applying OLS, we get the following plot of residuals: Applying the model to perform prediction of test data, we calculate the average of the absolute value of the errors:



```
abs(y_test - y_pred).mean()
```

The result is 0.314. This is not terrible, but not impressive either. With decision tree classifier, the same measure goes down to 0.194! There, we also experiment with maximum depth. We determine that the optimal maximum depth is 5. Lastly, we apply random forest classifier, it gets 0.213 as performance score specified above. Out of the three trials, decision tree classifier performed best.

We lastly did cross-validation with the `cross_validate` function. While there were no drastic performance differences in performance we got the following `maximum` test scores on the models:, `0.203`, `0.802` and `0.792`, respectively. This also verifies decision tree classifier performes best.