“Census Income” dataset modeling and analytics report

1 Problem Description

Facing the dataset named “Census Income Data Set”, our target is to predict whether the fin al income exceeds $50,000 per year or not based on features. We can easily discover that th e finally prediction target is a traditional classification problem and so, some approaches co uld be applied into this dataset to achieve it. In this project, what we expect of the final pred iction is to have over 80% of accuracy (comparing between prediction and test dataset) of t he income classification (over 50K per year or not).

In this report, we are going to analyze and display the result of prediction with the followin g structure: 1) Describe the original dataset and processing methods; 2) The issues we met d uring the project; 3) The approaches we utilized in this project(Theoretical and applicatio n); 4) The result of prediction and its related data or measurements; 5) Comparison of appro aches (Different models and ways of processing dataset); 6) Summary of the whole project and the conclusion.

2 Dataset Description

“Census Income Data Set” is a dataset which contains 48842 lines of items, 14 different attributes (Features, salary column not included) which are given by categories and integers. The descriptions of the features are as followed:

(1) age: the ages of people, numeric;
(2) workclass: type of working, category;
(3) fnlwgt: final weight in complete. It means the weights on the CPS files are controlled to independent estimates of the civilian noninstitutional population of the US, numeric;
(4) education: the final education of people, category;
(5) education-num: the number of education of people, numeric;
(6) marital-status: the marital status of people, category;
(7) occupation: area of job of people, category;
(8) relationship: status in the family, category;
(9) race: race of the people, category;
(10) sex: sex of the people, category;
(11) capital-gain: capital gain of people, numeric;
(12) capital-loss: capital loss of people, numeric;
(13) hours-per-week: hours to work per week of people, numeric;
(14) native-country: nationality of people.

When we starting to check about the content of the dataset, we can easily find that there’s some missing data in whether training dataset or testing dataset. After we changing all the “?” value existing in the dataset to “space” (“ ”), we can make an overview of the current dataset as Figure1.1 and Figure 1.2:

Besides, despite there are over 5% lines of the data have at least one feature missing, we decided to drop all the columns which have missing data.

Finally, this dataset is provided by the book “Proceedings of the Second International Conference on Knowledge Discovery and Data Mining”. (Kohavi, 1996)

3 Bottleneck

During the data analytics and target prediction, we met several difficulties, whether in the method of encoding data or the models chose to apply on the dataset.

According to the analytics towards the target prediction, it only has two results: “>50K” and “<=50K”, so we can define this dataset as a binary classification problem. Nevertheless, we have multiple choices for the model to train and fit data.

On the other hand, due to some specific data which are categorical. We found it makes some difficulties if we keep these columns of data as original, the accuracy of results will be lower and it will dissatisfy our expectation. So, we did mappings on categorical data and made them into numerical which can simplify the computing and modeling processes.

```
RangeIndex: 45842 entries, 0 to 45841
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             45842 non-null  int64
 1   workclass       43227 non-null  object
 2   final_wt        45842 non-null  int64
 3   education       45842 non-null  object
 4   education-num   45842 non-null  int64
 5   marital-status  45842 non-null  object
 6   occupation      43217 non-null  object
 7   relationship    45842 non-null  object
 8   race            45842 non-null  object
 9   sex             45842 non-null  object
 10  capital-gain    45842 non-null  int64
 11  capital-loss    45842 non-null  int64
 12  hours-per-week  45842 non-null  int64
 13  native-country  45033 non-null  object
 14  salary          45842 non-null  object
```

Fig.1.1. Information of training set

```
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             3000 non-null   int64
 1   workclass       2816 non-null   object
 2   final_wt        3000 non-null   int64
 3   education       3000 non-null   object
 4   education-num   3000 non-null   int64
 5   marital-status  3000 non-null   object
 6   occupation      2816 non-null   object
 7   relationship    3000 non-null   object
 8   race            3000 non-null   object
 9   sex             3000 non-null   object
 10  capital-gain    3000 non-null   int64
 11  capital-loss    3000 non-null   int64
 12  hours-per-week  3000 non-null   int64
 13  native-country  2952 non-null   object
 14  salary          3000 non-null   object
```

Fig.1.2. Information of testing set

4 Solution

4.1 Sigmoid Function

As it said in the third part, the target prediction values in this dataset are only divided into t wo categories. In this case, we can utilize the character of $Sigmoid$ function, which is defin ed as followed formula 1:

$$y = \frac{1}{1 + e^{-z}} \tag{1}$$

And its shape is depicted as figure 2:

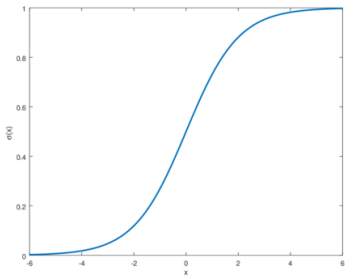

Fig. 2. Sigmoid Function

By using this function, it can help to classify the result into “>0.5” and “<0.5” and so we de cided to use Logistic Regression to be the first model.

4.2 Logistic Regression

Logistic Regression is specifically aiming to deal with binary classification problem because of the Sigmoid Function inside. Normally, the output of linear regression is showed in formula 2 and we can get the formula 3 of Logistic Regression by combining formula 1 and 2:

$$z = \omega^T x + b \tag{2}$$

$$y = f(x) = \frac{1}{1 + e^{-(-\omega^T x + b)}} \tag{3}$$

By utilizing this model, we can convert the result to a value close to 0 or 1. If the value is closer to 0, this model defines it as 0 and it is the same as the values which are closer to 1.

In this project, we decide to do two tests by using Logistic Regression. One is the basic model and another is with parameters.

4.3 Decision Tree

Because of the big amounts of features exist in this dataset, we decide to use Decision Tree to divide the columns (features) into several parts and layers to make the prediction be more precise. The structure of the Decision Tree in this case is presented in the file: tree_structure.txt.

4.4 Random Forest

A Random Forest is a meta estimator that fits a number of Decision Tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. (scikit-learn developers, 2022) We utilized random forest to build several decision trees to fit the dataset, which could have a higher fitting degree.

4.5 One-hot coding

Normally, learning models in machine learning expect the data in numeric but not strings, so we decided to use coding to the prediction column “income”. In this case, we know that there are only two categories, so it is better to use “One-hot” coding to compile the categorical values into numeric ones.

Here, the category of income “>50K” is translated as 1 and “<=50K” is translated as 0. And we can see the first 10 lines of data in the column “income” in the figure 3 as followed:

```
print(y_train.head(10))
```

```
0    0
1    0
2    0
3    0
4    0
5    0
6    0
7    1
8    1
9    1
Name: salary, dtype: int64
```

Fig. 3. One-hot coding result

4.6 Minmax Scaler:

To deal with the numerric data apart from “income”, we have to do the data normalization. Here, we utilized MinMaxScaler to map the data into [0, 1]. These procedure means we find the ratio of the distance between the elements in each column to the minimum value to the distance between the maximum value and the minimum value of the column, and so the range is in [0, 1] interval. After processing, the dataset will be presented as it showed in Figure 4.

	age	workclass	final_wt
0	0.301370	5	0.043350
1	0.452055	1	0.047274
2	0.287671	0	0.136877
3	0.493151	0	0.149792
4	0.150685	0	0.219998
...	...	...	...
45836	0.219178	0	0.156895
45837	0.301370	0	0.136723
45839	0.287671	0	0.244762
45840	0.369863	0	0.047666
45841	0.246575	2	0.114195

Fig. 4. MinMaxScaler data

5 Results

5.1 Logistic Regression (Original & Optimized)

First, without any parameters set, the model will have its default parameters. We tested two models and the interceptions and coefficients are displayed in Table 1:

Then I put some parameters to the Logistic Regression model. Which are shown in Table 2:

At last, we did predictions for the target values with the support of two models mentioned in this part and the results are showed in Figure 5.1 and 5.2:

Table 1. Interception and coefficient of two models

	Original Model	Optimized Model
Interception	[-7.36906025e-03, 1.82853780e-05, -3.22414695e-06, -1.36604507e-03, -1.73424311e-03 -1.50979110e-03, -3.73731473e-03 -1.45873086e-03, -1.75494836e-04, 2.01326926e-05, 3.39207884e-04, 7.79035286e-04, -8.47787585e-03, -1.32518273e-03]	[3.27545093e-02, 2.50374033e-02, 7.23462266e-07, 3.01906067e-02, 3.00970968e-01 -8.54123775e-01, -7.63037741e-02 -3.17091440e-01, -5.99427658e-02, 4.55661541e-01, 3.20690346e-04, 6.55377629e-04, 3.03779965e-02, -1.24879325e-02]
Coefficient	-0.00036503	-6.34427539

Table 2. Parameters of optimized Logistic Regression model

Parameter	Values
penalty	l1
solver	liblinear
tol	0.0001
C	1.0
fit_intercept	True
intercept_scaling	1

```
The data and report for normal model of Logistic Regression...

Score of the model:0.7900303822822885
              precision    recall  f1-score   support

           0       0.80      0.96      0.87      2075
           1       0.72      0.29      0.42       698

    accuracy                           0.79      2773
   macro avg       0.76      0.63      0.64      2773
weighted avg       0.78      0.79      0.76      2773
```

Fig. 5.1. Classification report of original model

```
The data and report for optimized model of Logistic Regression...

Score of the model:0.8396899361621033
              precision    recall  f1-score   support

           0       0.86      0.93      0.89      2075
           1       0.72      0.55      0.62       698

    accuracy                           0.83      2773
   macro avg       0.79      0.74      0.76      2773
weighted avg       0.83      0.83      0.83      2773
```

Fig. 5.2. Classification report of optimized model

5.2 Decision Tree & Random Forest

Then comes to the models of Decision Tree and Random Forest. Here are the parameters we utilized to build these two models, which are shown in Table 3.1 and 3.2:

Then we did the prediction as usual, then print the classification reports which are shown in Figure 6.1 and 6.2:

Table 3.1. Parameters of Decision Tree model

Parameters	Values
criterion	entropy
max_depth	5
min_samples_split	15
random_state	4

Table 3.2. Parameters of Random Forest model

Parameters	Values
n_estimators	10
max_depth	10
min_simples_split	15
max_features	sqrt

```
The data and report for Decision Tree...

Score of the model:0.8445427762748743
              precision    recall  f1-score   support

           0       0.86      0.95      0.90      2075
           1       0.77      0.52      0.62       698

    accuracy                           0.84      2773
   macro avg       0.81      0.74      0.76      2773
weighted avg       0.83      0.84      0.83      2773
```

Fig. 6.1. Classification report of Decision Tree

```
The data and report for Random Forest...

Score of the model:0.8559211983747831
              precision    recall  f1-score   support

           0       0.86      0.93      0.89      2075
           1       0.72      0.55      0.62       698

    accuracy                           0.83      2773
   macro avg       0.79      0.74      0.76      2773
weighted avg       0.82      0.83      0.82      2773
```

Fig. 6.2. Classification report of Random Forest

6 Analytics

As we can see from the analytics in chapter 5, the models all fit not bad on the dataset, they all have at least around 80% of the accuracy.

We compared some data being calculated by utilizing the prediction we got. The measurements are: 1) accuracy_score; 2) precision_score; 3) recall_score; 4) f1_score, and the results are shown in Table 3:

In General, we can observe that the degree of model fitting strengthens gradually. Except the precision_score of Decision Tree stands the top (slightly higher than Random Forest), for the other three measurements, Random Forest has best figures. So according to the scores we got from the fitting between prediction and test data, we can say that the **Random Forest model fits data the best**.

Besides, I calculated the distribution of the number and ratio of people's income (for original data), which is displayed in Figure 7.

Table 3. Measurements of four models fitting

	Original LR	Optimized LR	Decision Tree	Random Forest
accuracy_score	0.7937	0.832	0.8406	0.8464
precision_score	0.725	0.7164	0.77	0.7519
recall_score	0.2908	0.5501	0.5229	0.5817
f1_score	0.4151	0.6224	0.6229	0.6559

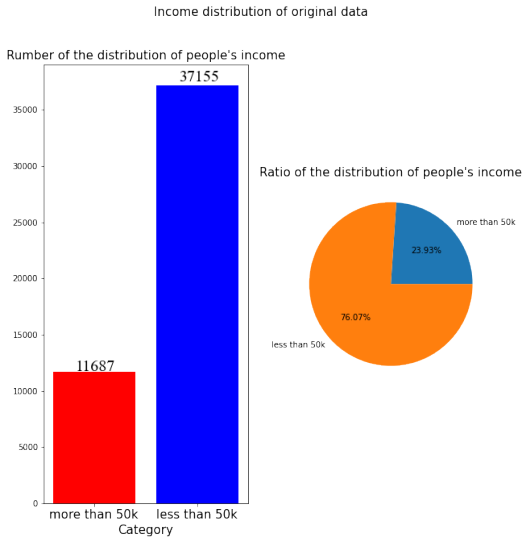

Fig. 7. Income distribution of original data

And we calculated the distribution of the number and ratio of people's income for all the mo dels we mentioned in this report and try to find the common feature of them. The data of mo dels are displayed in Table 4:

Table 4. Ratio of the distribution of people's income (four models)

	Original LR	Optimized LR	Decision Tree	Random Forest	Test dataset
More than 50K	10.10%	19.33%	17.09%	19.47%	25.17%
Less than 50K	89.90%	80.67%	82.91%	80.53%	74.83%

According to the figures showed in table 4, we can clearly see that the prediction given by u tilizing model of Random Forest has closer ratio to the test dataset while the Logistic Regre ssion without parameter performed the worst.

7 Conclusion

In this project, we proposed four models to fit the dataset given. The target is to classify wh ether a person's income is more than 50K or less than 50K according to 14 features. In this dataset, we used one-hot encoding and mappings to replace categoric data. ("income" colum n for one-hot encoding and the others for mappings respectively)

First, we utilized Logistic Regression (without parameters and optimized model) to fit datas et as this dataset belongs to a binary classification problem, the sigmoid function inside Log istic Regression helps to classify the prediction. Besides, we utilized Decision Trees and Ra ndom Forest, first to divide features into several groups, then build several Decision Trees t o contribute to a Random Forest, which can help significantly the result.

As we compared four measurements of classification problems, we observed that Random F orest fits the best to this dataset, then is Decision Tree and the worst is Logistic Regression, but it still has an accuracy of around 80%, which achieved our aim in this project.

However, the models proposed in this report are not over 95% of fitting. So, in the future st udy, we will keep studying new models and give them more appropriate parameters to fit th e dataset better.

Bibliography

Kohavi, R. (1996). *Proceedings of the Second International Conference on Knowledge Disco very and Data Mining.*

scikit-learn developers. (2022). *scikit-learn-RandomForestClassifier.* Retrieved from Rando mForestClassifier: https://scikit-learn.org/stable/_sources/modules/generated/sklearn.en semble.RandomForestClassifier.rst.txt