Readme

Background:

The dataset comes from the 1994 Census database. The prediction task is to determine whether a person makes over 50K a year.

Here are a couple of original files I've got:

- trainFeatures.csv-----the original train data which contains 34189 samples totally.
- trainLabels.csv----the label file which contains 34189 labels.
- testFeatures.csv---the original test data which need to be predict and it doesn't have any label previously, the samples are 14653.

After checking the data, I find that the data, on the whole, are strings that do not contain missing values, both training and testing. It means that I have to process those string data before training the model.

Data Analysis

All we know, before the Data Mining projects, we have to do dome exploratory data analysis. The analysis steps and analysis code have been saved, named A2_hchenbw_12233738_Data Analysis.ipynb in the folder.

At this step, I find that some variables like education-num, hours-per-week are categorical and are numbers. I don't have to do more process on the variables like them. As some variables like age and fnlwgt, although they're categorical and numerical, in order to fit the model better, it would be better if we transform them into another space. (e.g. age, 0~18 is equal to 1, 18~30 is equal to 2). For the rest of the variables, we just need to encode them because we have changed the string to number.

Part1:

Please check out the jupyter notebook file in the root folder named A2_hchenbw_12233738_Data Analysis.ipynb. The file describing how to process variables and some details.

==========

Part2:

model:

This is the model of the project. Here are my steps:

step1->process the features

step2->split the training feature set, 80% for train, 20% for test

step2->choose model, here I use Random Forests, and optimize the model

step3->train the model, save model, named 'RandomForestClassifier.pkl'

step4->make prediction

step5->save the predict result, the file named 'prediction-result.csv'

For the details of how I train the model, through cross validation (5 folds):

K-fold:1, train samples=21880, test samples=5471, train accuracy score=0.9037 test accuracy score is=0.8658

K-fold:2, train samples=21881, test samples=5470, train accuracy score=0.8947 test accuracy score is=0.902

K-fold:3, train samples=21881, test samples=5470, train accuracy score=0.8945 test accuracy score is=0.9027

K-fold:4, train samples=21881, test samples=5470, train accuracy score=0.8926 test accuracy score is=0.9102

K-fold:5, train samples=21881, test samples=5470, train accuracy score=0.8952 test accuracy score is=0.9

mean test accuracy on 5 K-folds=0.8961400000000002

accuracy on complete test data is 0.8677

==========

How to run model: just run the model file: A2_hchenbw_12233738_code_model.py.

You may also put this python script into the IDE Spyder, it'll work as above.

Feature Engineering

Processing variables we have: except variable age, fnlwgt education-num and hours-per-week, I encode the rest variables to categorical numerical, e.g. variable ‘native-country’ and its value ‘United-States’ encode as 1, ‘Mexico’ encode as 2, etc.

For the age, fnlwgt, education-num and house-peer-week, please check the DataAnalysis.ipynb file.

As new features, here I created a new variable named ‘diff’ where diff = ‘capital-gain’-’capital-loss’.

Model

As we all know, ensemble models perform better than others. Here I use Random Forests which is a three-ensemble model. It is often doing well in the classification problem

For the task, I use accuracy score to evaluate my model, my accuracy score is 0.86+

Here are my model parameters:

Parameter name	Value
n_jobs	-1
n_estimators	160
min_samples_split	4
min_samples_leaf	1
max_features	auto(auto equal to sqrt(feature nums))
warm_start	True
max_leaf_nodes	None
bootstrap	True
random_state	1