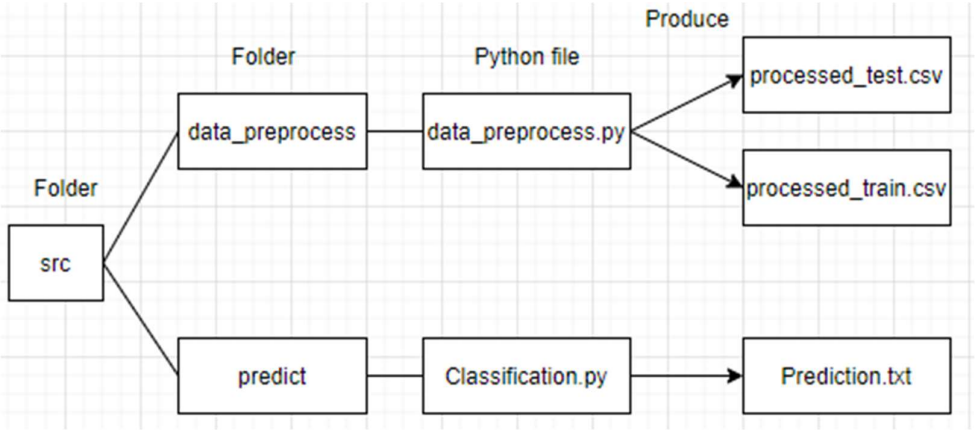
Miner2 username : Bear
Mason user id : chsiung2
Gnumber : G01272835
Best public score : 0.60

1. Introduction

- Data preprocess : I use sklearn.preprocessing.LabelEncoder to handle different attribute type : categorical and numerical and different data type : string and integer.
- The following picture is the structure of my src folder :
  i. "data_preprocess.py" : handle training and test data and the create two ".csv" file.
  ii. "Classification.py" : Use different classification models to get predictions. After executing the program, decide whether to drop F5 & F6 columns and then input a number (1~5) to choose a model. There are five models : Logistic Regression, Perception, Naïve Bayes, Support Vector Machine, Decision Tree.
  If Decision Tree was picked, it needs two more input to decide its arguments.



2. Approaches

- Handling "test.csv" and "train.csv":
  i. F10、F11 : use "fit_transform()" function in LabelEncoder. It will convert string categorical value into integer. Convert [' Amer-Indian-Eskimo' ' Asian-Pac-Islander' ' Black' ' Other' ' White'] to [0, 1, 2, 3, 4], [' female' 'male'] to [0, 1]. Because both train and test data contain the same number of categories, the rule of transformation is also the same. If test data lack for 'Other', the labeling rule would be different from train data.
  ii. id : drop this column because it can't offer any useful information.
  iii. F1、F2 : compute their quartiles first, and then use quartiles to classify the continuous values. I use function "pandas.cut()" to classify and label data. Test data use the quartiles of training data to make sure they are classified in the same way.
  iv. F5、F6 : these two data are too sparse to use quartile. So, I split the range of their value into four and then use "pandas.cut()" to classify and label them. Ex : (-1,0], (0,15000],(15000,30000], (30000,∞]

- Prediction :
  i. Use the modules in sklearn:
     1. Logistic Regression
     2. Perceptron
     3. Naive Bayes
     4. Support Vector Machine
     5. Decision Tree : there are two argument need to be decided.
        - Criterion : gini index or entropy
        - max_depth : the max depth of this tree

3. Experimental Results

- Logistic Regression : 0.39
- Perception : 0.48
- Naive Bayes : 0.54
- Support Vector Machine : 0.56
- Decision Tree :
  - Gini index : 0.49
  - Entropy : 0.51
  - Entropy、Max depth of tree = 3 : 0.58
  - Entropy、Max depth of tree = 5 : 0.58
  - Entropy、Max depth of tree = 11 : 0.58
  - Entropy、Max depth of tree = 11、**Get rid of F5&F6** : 0.60

4. Conclusion

- Decision Tree with entropy got the higher grade than with gini index. Also, the max depth : 11 was better than others with this test data.
- Because F5、F6 have lots of 0 data, only few have value, I got my highest score 0.60 after getting rid of these two columns.
- I used quartiles to handle continuous value like F1 and F2. But I encountered a problem that two of them would be the same. Ex : Q1 = Q2 = 30. Therefore, I use quantile(0.2) or quantile(0.6) to make data classified successfully. It will decrease the precision in some degree.