Assignment 2

COURSE: PRACTICAL DATA SCIENCE

HUYNH AI LOAN – s3655461

BINH CHON NHUT LE - s3256292

Table of Contents

Abstract

The report covers the whole process of Data Science including Data Preprocessing, Data Exploration and Data Modelling, in which Data Modelling is the main part. The 1994 US Census dataset is used to build classifiers to predict an individual earn money more than 50,000 per year. Three classifiers KNN, RandomForest and Decision Tree are used to build model from this dataset. The 5-folds cross validation is used to validate the result of each model and allows us the discover which model might produce better result than the other ones.

Introduction

This objective of this project is to build classifiers to predict whether an individual has income more than or less than 50K per year based on the 1994 US Census dataset. The dataset is collected via UIC website . This report is organized into six sections. Section I describes data set. Section II explores which attributes might be helpful to predict problem. The overview of our methodology is mentioned in section III. The detail of performance analysis and comparison between methods will be discussed in Section IV and V. The last part concludes with a summary.

Dataset

“Adult” dataset provided by UIC has total 48841 records divided into two datasets “adult.data” and “adult.test”. There are 32,560 training observations in “adult.data” training dataset, and 16,281 test observations in “adult.test”. Both datasets consist of 14 descriptive features and 1 target feature. In this project, we combine two these datasets into one.

Data description

Descriptive features

Attribute	Description
Age	Continuous
Workclass	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
fnlwgt	Continuous
Education	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
Education-num	Continuous
Marital-status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
Occupation	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
Relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
Race	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
Sex	Female, Male
Capital-loss	Continuous
Capital-gain	Continuous
Hours-per-week	Continuous
Native-country	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Target feature

Income >50K , Income <=50K

Data Preparation and Data Exploration

Data Preparation

This dataset has no null value. However, it got some typing issues. In this report, we only describe some main attributes in this step

Workclass : is divided into four groups: Private, Gov, Self-emp and Other

Education: is categorized into groups which are Post graduate, Associate, Before High School, Some-college, Bachelors and High school graduate

Marital-Status: is grouped into 5 categories consisting of Married, Never-married, Divorced, Separated, Widowed.

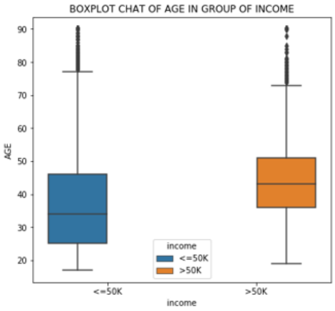
Native country: because United-States takes the majority of dataset, therefore we divide this attribute into two groups: United-States and Non-US.

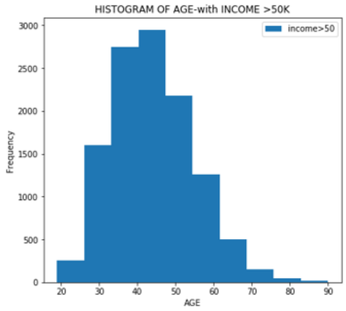
About code to do cleaning, please refer in Data Preprocessing notebook.

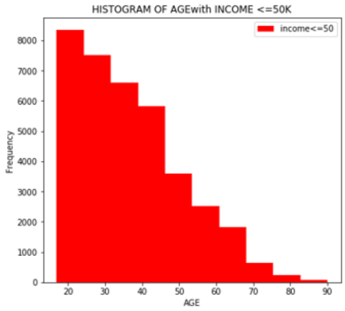
Data Exploration

In this part, we only summarize why we choose useful features in predicting Income class.

Age

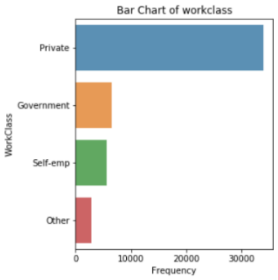


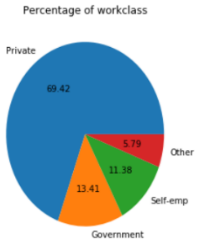


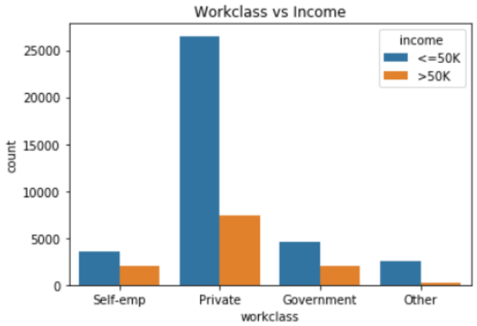


It is obviously to see that individuals with age between 30-45 earn more than 50K. The majority of people with age less than 30 earn less money. Hence Age attribute may be useful feature affecting to target class.

Workclass:

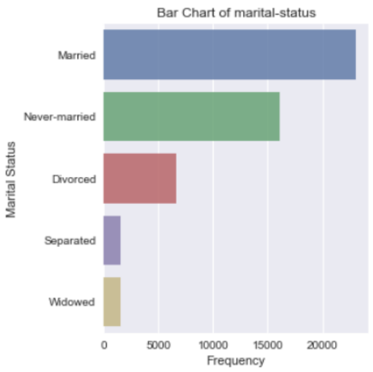


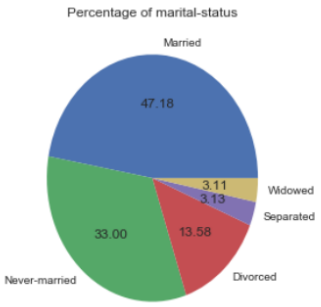


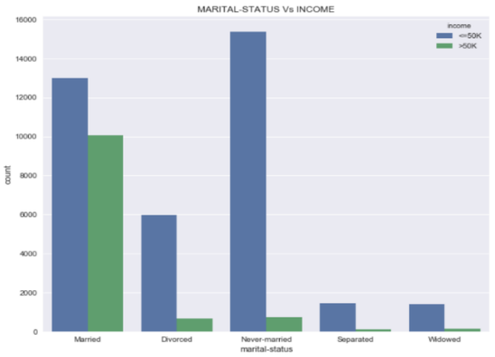


In general, regardless of workclass, the majority of people got income less than 50K. However, the private workclass earn more money rather other classes.

Marital Status

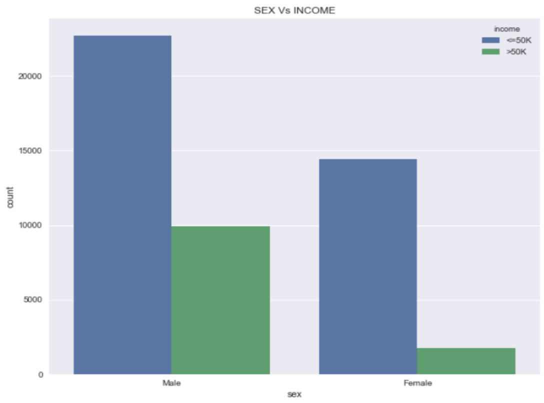






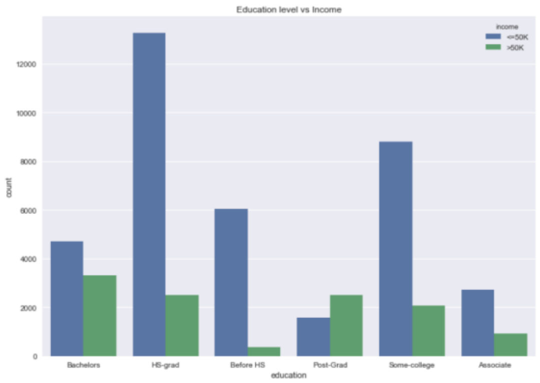
The rate of never married people earns money rather than the other groups. However, the number of married people with income larger than 50,000 is the highest. Therefore, the marital status may provide helpful information in prediction.

Sex



Obviously, Male earns more money than Female.

Education:



It’s clearly to see that in most of education levels except Post graduate, the number of people having income less than 50,000 was always higher than those with income more than 50,000. In fact, the majority of people with only high school graduate earned money less than 50,000.

In general, from the data exploration, we found that five attributes including marital status, sex, age, workclass and education level were potentially useful features in predicting Income class.

Methodology

As mentioned in Data Exploration part, there are only five attributes which may be provide useful information in predicting target class. We found that the majority of people with age between 30-45 earn income more than 50K. Therefore, we categorized age in 4 groups which are <=30, 30-45, 45-60, >60. Since these attributes are categorical variables, then we considered to use Decision Tree (DT) and RandomForest (RF) classifier for Data Modelling. However, we also do experiment with K-Nearest Neighbor (KNN) to see whether KNN is suitable model for this dataset or not.

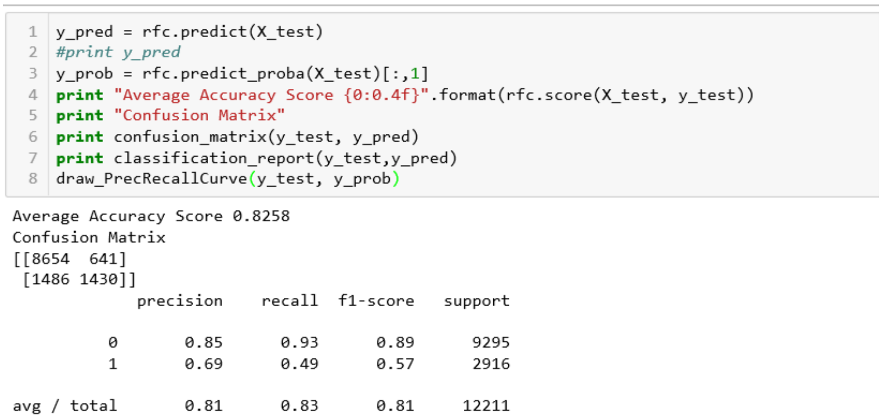
RandomForest is an ensemble learning which can be considered as extension of Decision Tree. It will build many sub-trees based on features and sample selected randomly. The advantage of RF is dealing with overfitting problem. That is reason why we choose two these classifiers in order to explore which classifier is better than the other one. For RF, we do training a random forest from 10 decision trees and use entropy criterion as an impurity measure to split the nodes.

In this experiment, we use Cross Validation with 5 folds to validate the result of models.

Results

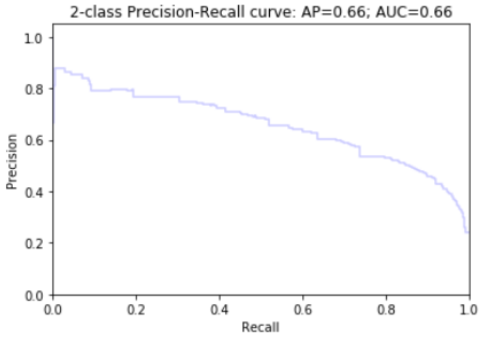
RandomForest Classifier:

This is the result of RandomForest classifier with n_estimators = 10, max_features=5

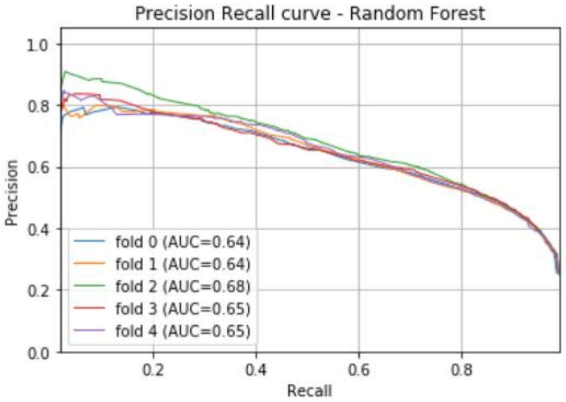


```
y_pred = rfc.predict(X_test)
#print y_pred
y_prob = rfc.predict_proba(X_test)[:,1]
print "Average Accuracy Score {0:0.4f}".format(rfc.score(X_test, y_test))
print "Confusion Matrix"
print confusion_matrix(y_test, y_pred)
print classification_report(y_test,y_pred)
draw_PrecRecallCurve(y_test, y_prob)
```

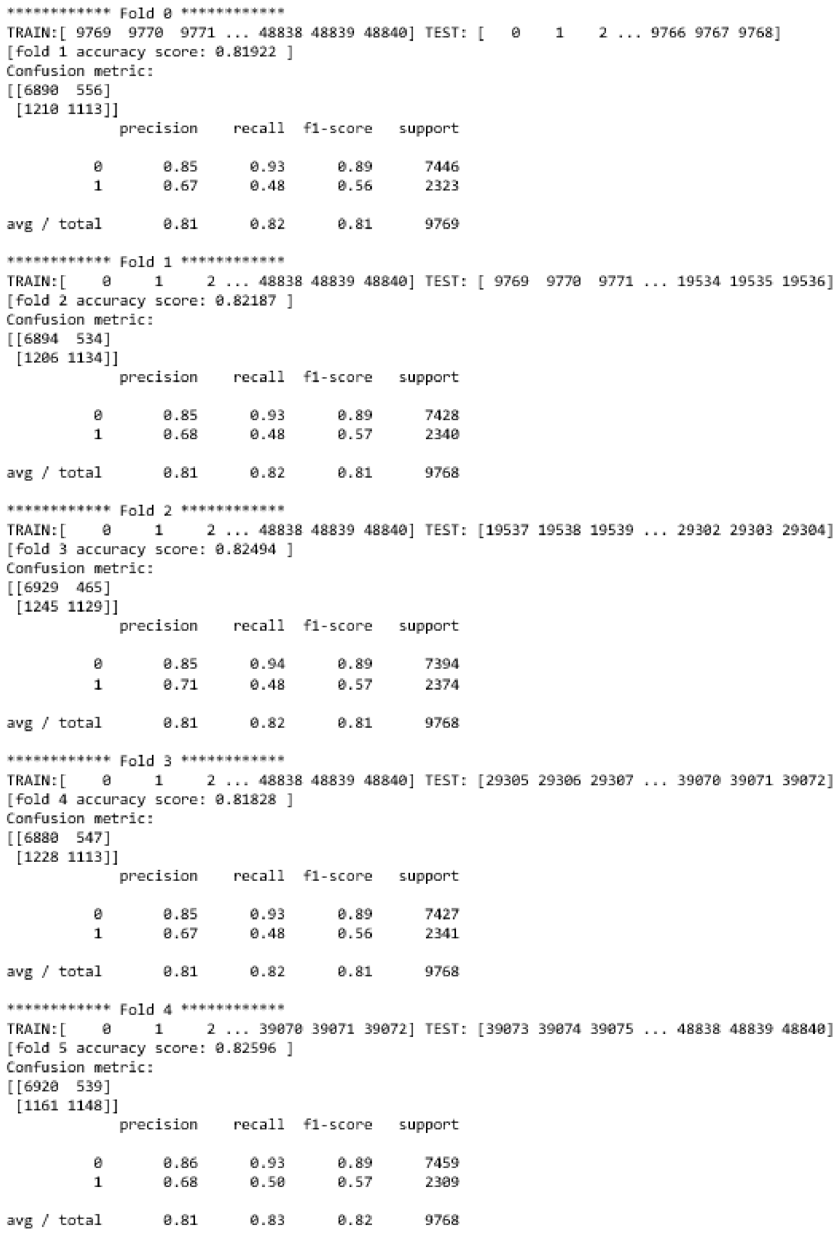
```
Average Accuracy Score 0.8258
Confusion Matrix
[[8654  641]
 [1486 1430]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      9295
          1       0.69      0.49      0.57      2916

avg / total       0.81      0.83      0.81     12211
```



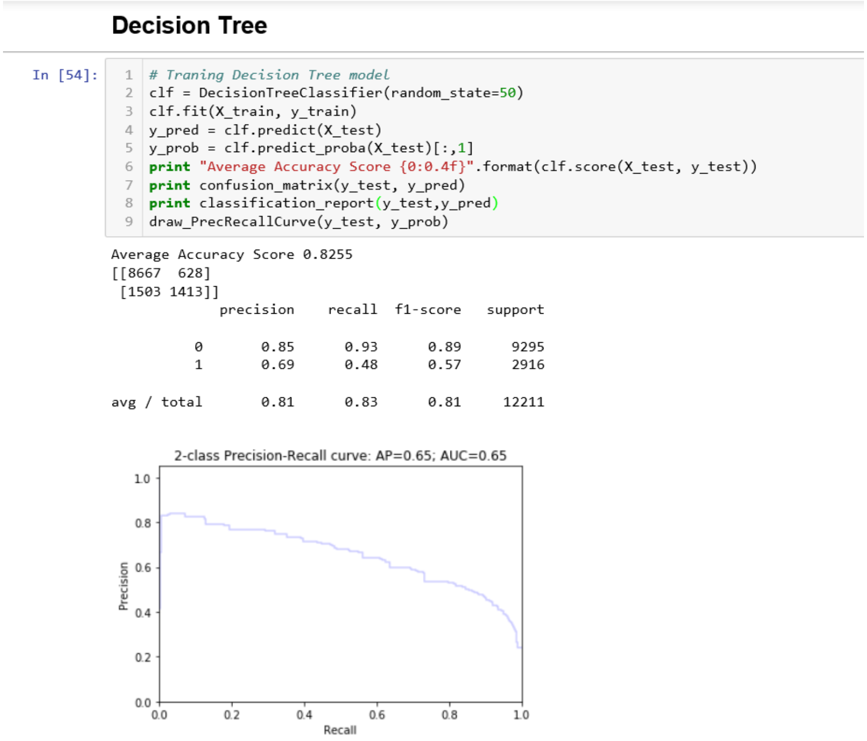
Precision Recall for 5-folds cross validation:



5-folds cross validation result:


```
************ Fold 0 ************
TRAIN:[ 9769  9770  9771 ... 48838 48839 48840] TEST: [    0     1     2 ... 9766 9767 9768]
[fold 1 accuracy score: 0.81922 ]
Confusion metric:
[[6890  556]
 [1210 1113]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      7446
          1       0.67      0.48      0.56      2323

avg / total       0.81      0.82      0.81      9769

************ Fold 1 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [ 9769  9770  9771 ... 19534 19535 19536]
[fold 2 accuracy score: 0.82187 ]
Confusion metric:
[[6894  534]
 [1206 1134]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      7428
          1       0.68      0.48      0.57      2340

avg / total       0.81      0.82      0.81      9768

************ Fold 2 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [19537 19538 19539 ... 29302 29303 29304]
[fold 3 accuracy score: 0.82494 ]
Confusion metric:
[[6929  465]
 [1245 1129]]
             precision    recall  f1-score   support

          0       0.85      0.94      0.89      7394
          1       0.71      0.48      0.57      2374

avg / total       0.81      0.82      0.81      9768

************ Fold 3 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [29305 29306 29307 ... 39070 39071 39072]
[fold 4 accuracy score: 0.81828 ]
Confusion metric:
[[6880  547]
 [1228 1113]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      7427
          1       0.67      0.48      0.56      2341

avg / total       0.81      0.82      0.81      9768

************ Fold 4 ************
TRAIN:[    0     1     2 ... 39070 39071 39072] TEST: [39073 39074 39075 ... 48838 48839 48840]
[fold 5 accuracy score: 0.82596 ]
Confusion metric:
[[6920  539]
 [1161 1148]]
             precision    recall  f1-score   support

          0       0.86      0.93      0.89      7459
          1       0.68      0.50      0.57      2309

avg / total       0.81      0.83      0.82      9768
```

Decision Tree



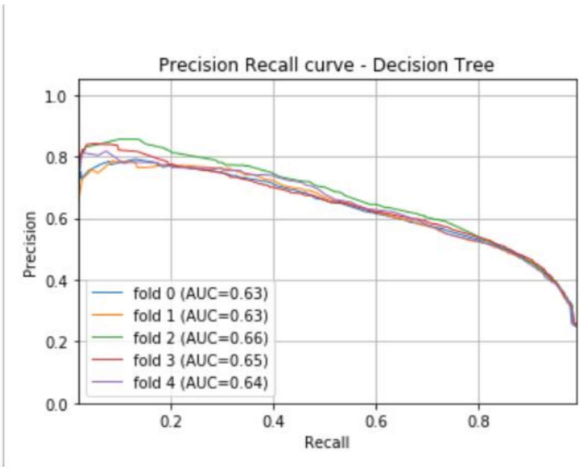
Decision Tree

In [54]:

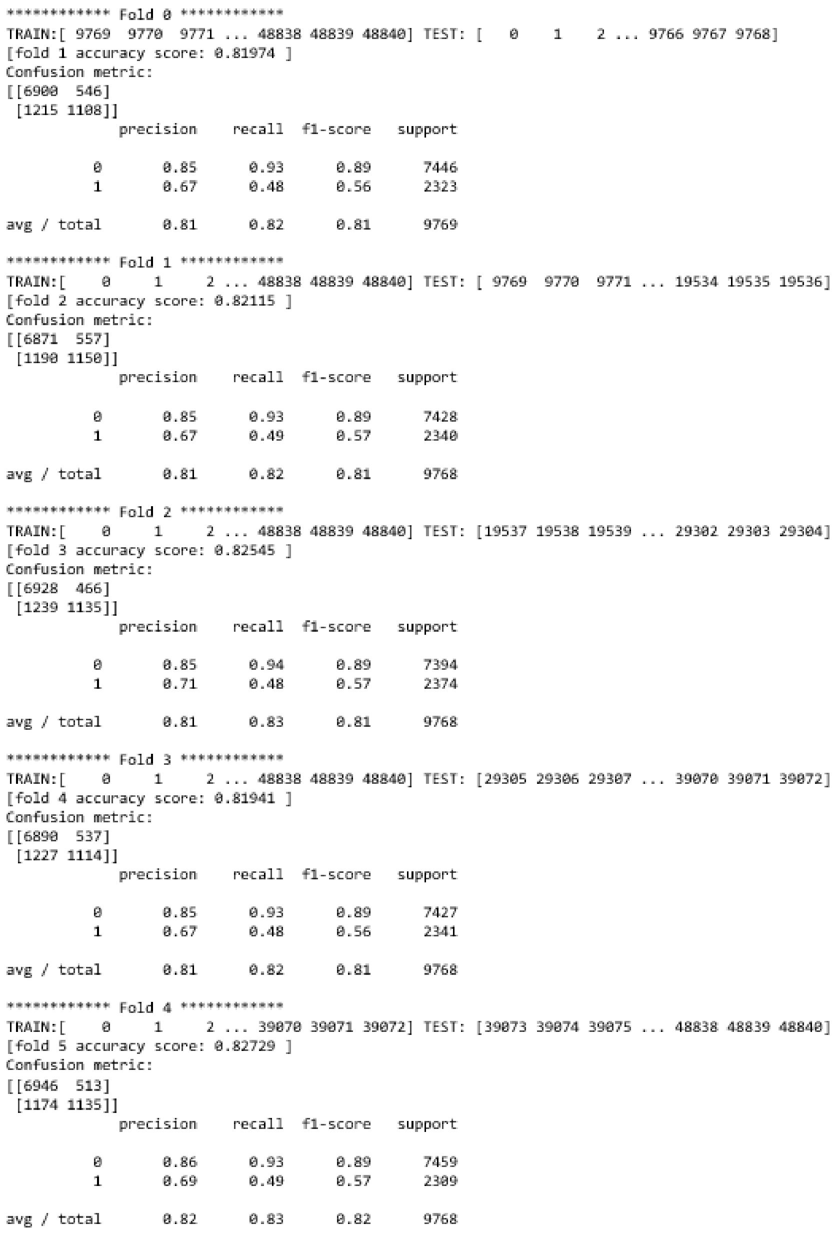
```
# Traning Decision Tree model
clf = DecisionTreeClassifier(random_state=50)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
y_prob = clf.predict_proba(X_test)[:,1]
print "Average Accuracy Score {0:0.4f}".format(clf.score(X_test, y_test))
print confusion_matrix(y_test, y_pred)
print classification_report(y_test,y_pred)
draw_PrecRecallCurve(y_test, y_prob)
```

```
Average Accuracy Score 0.8255
[[8667  628]
 [1503 1413]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      9295
          1       0.69      0.48      0.57      2916

avg / total       0.81      0.83      0.81     12211
```

Precision Recall curve for 5-folds cross validation:



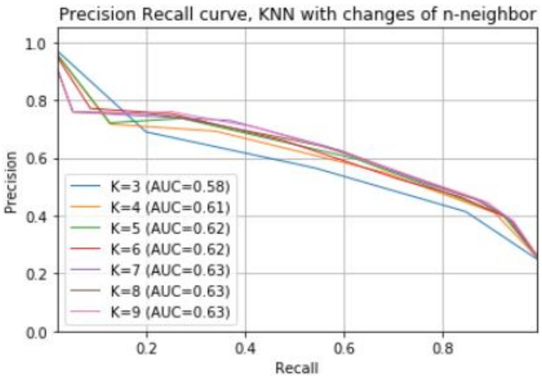
5-Fold cross validation result:



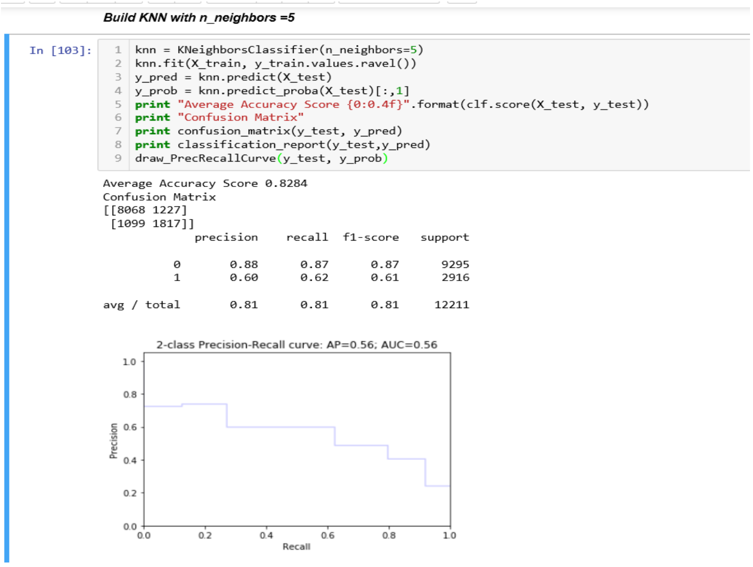
```
************ Fold 0 ************
TRAIN:[ 9769  9770  9771 ... 48838 48839 48840] TEST: [   0    1    2 ... 9766 9767 9768]
[fold 1 accuracy score: 0.81974 ]
Confusion metric:
[[6900  546]
 [1215 1108]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      7446
          1       0.67      0.48      0.56      2323

avg / total       0.81      0.82      0.81      9769

************ Fold 1 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [ 9769  9770  9771 ... 19534 19535 19536]
[fold 2 accuracy score: 0.82115 ]
Confusion metric:
[[6871  557]
 [1190 1150]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      7428
          1       0.67      0.49      0.57      2340

avg / total       0.81      0.82      0.81      9768

************ Fold 2 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [19537 19538 19539 ... 29302 29303 29304]
[fold 3 accuracy score: 0.82545 ]
Confusion metric:
[[6928  466]
 [1239 1135]]
             precision    recall  f1-score   support

          0       0.85      0.94      0.89      7394
          1       0.71      0.48      0.57      2374

avg / total       0.81      0.83      0.81      9768

************ Fold 3 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [29305 29306 29307 ... 39070 39071 39072]
[fold 4 accuracy score: 0.81941 ]
Confusion metric:
[[6890  537]
 [1227 1114]]
             precision    recall  f1-score   support

          0       0.85      0.93      0.89      7427
          1       0.67      0.48      0.56      2341

avg / total       0.81      0.82      0.81      9768

************ Fold 4 ************
TRAIN:[    0     1     2 ... 39070 39071 39072] TEST: [39073 39074 39075 ... 48838 48839 48840]
[fold 5 accuracy score: 0.82729 ]
Confusion metric:
[[6946  513]
 [1174 1135]]
             precision    recall  f1-score   support

          0       0.86      0.93      0.89      7459
          1       0.69      0.49      0.57      2309

avg / total       0.82      0.83      0.82      9768
```

KNN

Building KNN models with n_neighbors from 3 to 10, the precision recall curve is stated as below:



From the above figure, it’s obviously to see that if from k=5, the area under the precision-recall curve (AUC) only changes a little bit. Therefore, we will test this model with k=5. The result is identified as below:



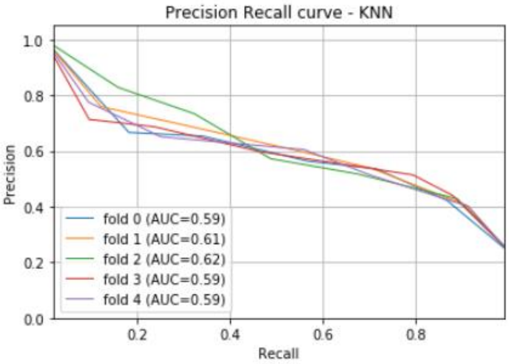
Build KNN with n_neighbors =5

In [103]:

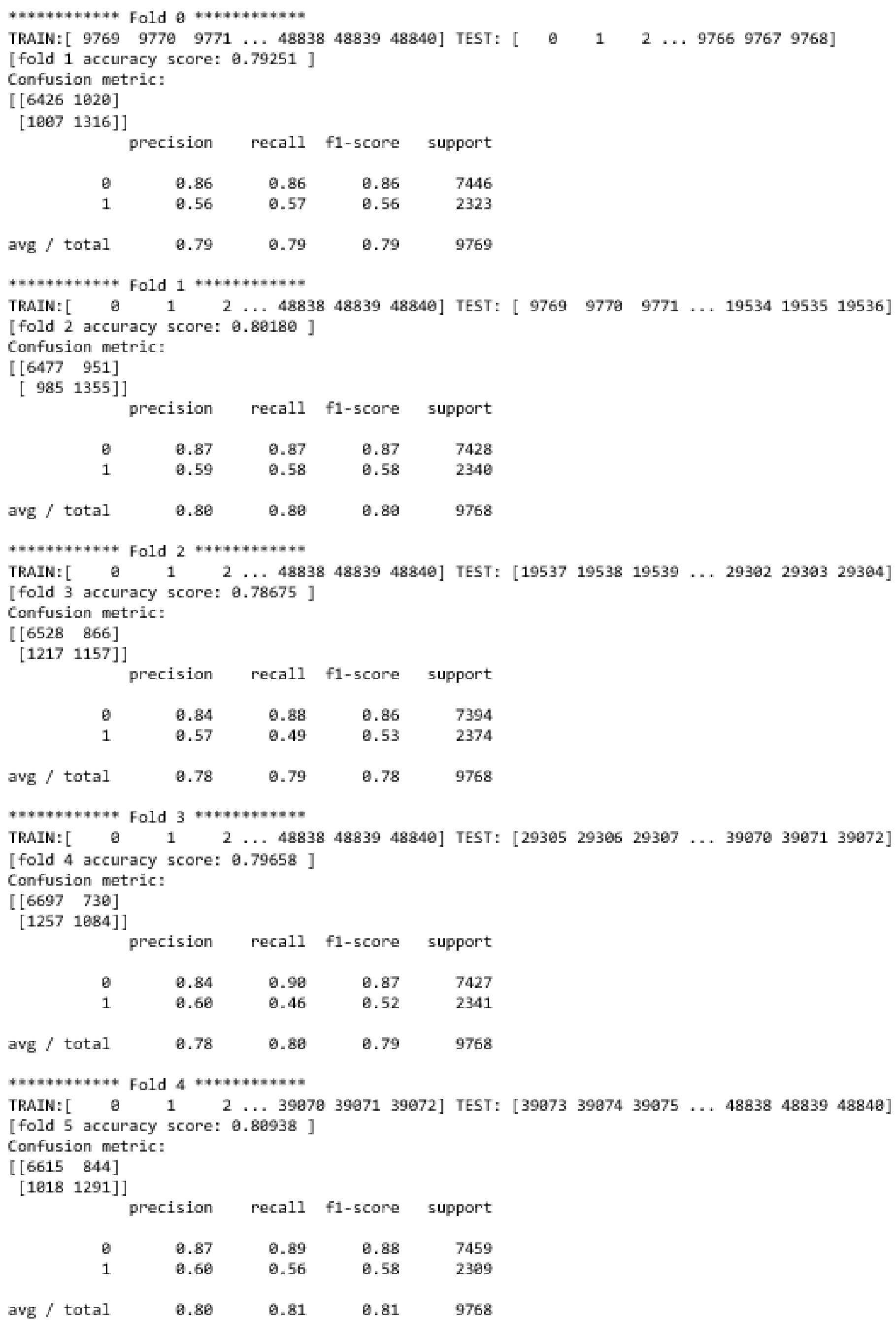
```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train.values.ravel())
y_pred = knn.predict(X_test)
y_prob = knn.predict_proba(X_test)[:,1]
print "Average Accuracy Score {0:0.4f}".format(clf.score(X_test, y_test))
print "Confusion Matrix"
print confusion_matrix(y_test, y_pred)
print classification_report(y_test,y_pred)
draw_PrecRecallCurve(y_test, y_prob)
```

```
Average Accuracy Score 0.8284
Confusion Matrix
[[8068 1227]
 [1099 1817]]
             precision    recall  f1-score   support

          0       0.88      0.87      0.87      9295
          1       0.60      0.62      0.61      2916

avg / total       0.81      0.81      0.81     12211
```

Precision recall curve with 5-folds cross validation:



5-folds cross validation result:



```
************ Fold 0 ************
TRAIN:[ 9769  9770  9771 ... 48838 48839 48840] TEST: [   0    1    2 ... 9766 9767 9768]
[fold 1 accuracy score: 0.79251 ]
Confusion metric:
[[6426 1020]
 [1007 1316]]
             precision    recall  f1-score   support

          0       0.86      0.86      0.86      7446
          1       0.56      0.57      0.56      2323

avg / total       0.79      0.79      0.79      9769

************ Fold 1 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [ 9769  9770  9771 ... 19534 19535 19536]
[fold 2 accuracy score: 0.80180 ]
Confusion metric:
[[6477  951]
 [ 985 1355]]
             precision    recall  f1-score   support

          0       0.87      0.87      0.87      7428
          1       0.59      0.58      0.58      2340

avg / total       0.80      0.80      0.80      9768

************ Fold 2 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [19537 19538 19539 ... 29302 29303 29304]
[fold 3 accuracy score: 0.78675 ]
Confusion metric:
[[6528  866]
 [1217 1157]]
             precision    recall  f1-score   support

          0       0.84      0.88      0.86      7394
          1       0.57      0.49      0.53      2374

avg / total       0.78      0.79      0.78      9768

************ Fold 3 ************
TRAIN:[    0     1     2 ... 48838 48839 48840] TEST: [29305 29306 29307 ... 39070 39071 39072]
[fold 4 accuracy score: 0.79658 ]
Confusion metric:
[[6697  730]
 [1257 1084]]
             precision    recall  f1-score   support

          0       0.84      0.90      0.87      7427
          1       0.60      0.46      0.52      2341

avg / total       0.78      0.80      0.79      9768

************ Fold 4 ************
TRAIN:[    0     1     2 ... 39070 39071 39072] TEST: [39073 39074 39075 ... 48838 48839 48840]
[fold 5 accuracy score: 0.80938 ]
Confusion metric:
[[6615  844]
 [1018 1291]]
             precision    recall  f1-score   support

          0       0.87      0.89      0.88      7459
          1       0.60      0.56      0.58      2309

avg / total       0.80      0.81      0.81      9768
```

Discussion

Following the results, we found that RandomForest gave better results than two other models. We did test RF with changes of n_estimators parameter; the average accuracy score, precision, recall and f1 score were changed depending on value of n_estimators. It might imply that RF can deal with overfitting problem rather than Decision Tree and improve the accuracy better than DT.

It is clearly to show that KNN might not be appropriate given there were many categorical features in the data (for example, this dataset).

Conclusion

In summary, the Random Forest produces the best performance in predicting whether individual earns more than 50K per year. We use 5-folds cross validation to validate and compare the results between classifiers. For future works, we may consider how to deal with imbalance class issue in this dataset to give the better result in prediction.

Reference

Kelleher, J., MacNamee, B. and D'Arcy, A. (2015). “Fundamentals of machine learning for predictive data analytics”. Cambridge, Mass.: MIT Press.