Distinction Task 10.1D: Model evaluation metrics

Task description:

The model evaluation and selection techniques are the most important tools in a data scientist's toolbox. So far, we have introduced many model evaluation methods/metrics, such as GridSearchCV, cross_val_score, confusion matrix, precision, recall and f-score, etc. In reality, classification problems rarely have balanced classes, and often false positives and false negatives have very different consequences. We need to understand what these consequences are, and pick an evaluation metric accordingly, therefore select a right model for the given dataset.

In this task, you are given a dataset "creditcard.csv" used in practical10. Based on the code example provided in practical10, try to find the "best" classification model by comparing the evaluation metrics, especially the recall rates produced by knn, decision tree and random forest models.

You are given:

- Dataset: creditcard.csv
- thresholds = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9]
- Parameter grid (param_grid):
  For knn, n_neighbors = [1, 2, 3, 4, 5]
  For decision tree, max_depth = [3, 4, 5, 6, 7]
  For random forest, n_estimators = [5, 10, 20, 50]
- GridSearchCV(model_classifier(random_state=0), {param: param_grid}, cv=5, scoring='recall')
- Other parameters of your setting

You are asked to:

- use the train and test sets split in practical10 (X_train, X_test, y_train, y_test, and X_train_undersample, X_test_undersample, y_train_undersample, y_test_undersample)
- use Grid search with cross-validation to fit the undersample data with model knn, decision tree and random forest, respectively, set cv=5
- find and print the best parameter for each model (knn, decision tree or random forest) for X_train_undersample dataset
- for each model, build classifier using the found best parameter, predict using test sets (X_test_undersample and X_test), and plot the confusion matrix for the two predictions.
- for each model, plot recall matric for different threshold for the undersample dataset
- for each model, plot precision-recall curve for the undersample dataset

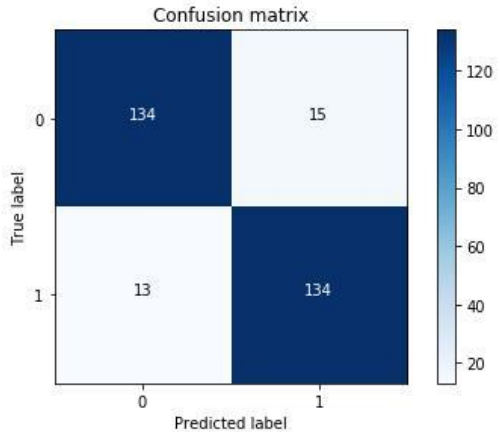
Note: It is very likely you will find the best parameters found for undersample dataset do not work well for the whole skewed dataset, which is normal. The ideal solution is to use GridSearchCV to find the best parameters for the whole skewed dataset, then use the best parameters to build a new classifier for the whole skewed dataset, however it takes TOO LONG on an office/home laptop/computer due to the size of the whole skewed dataset and amount of resources required. If

conditions allow, you are recommended to have a try. In this task, we will mainly play with the undersample dataset.

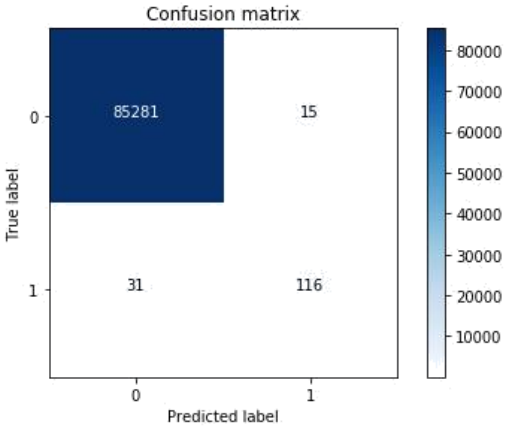
It is also recommended you define functions for searching best parameters, plotting curves/matrices, etc. as each model will be using similar code to produce the output.

Sample output as shown in the following are **for demonstration purposes only**. Yours might be different from the provided.

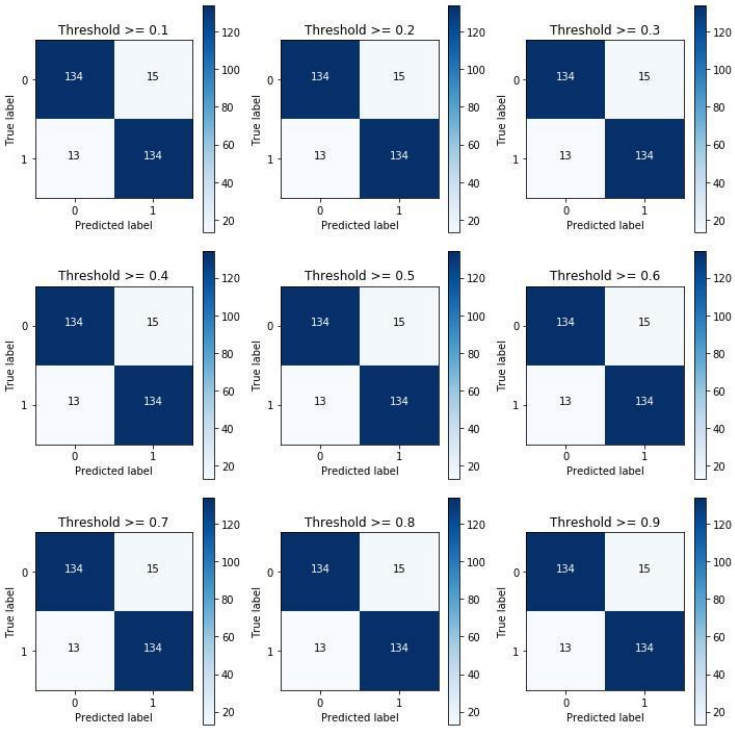
```
Percentage of normal transactions:  0.5
Percentage of fraud transactions:  0.5
Total number of transactions in resampled data:  984
Number transactions whole train dataset:  199364
Number transactions whole test dataset:  85443
Total number of whole transactions:  284807

Number transactions undersample train dataset:  688
Number transactions undersample test dataset:  296
Total number of undersample transactions:  984


Knn:


Best parameters set found on development set:
{'n_neighbors': 1}

Grid scores on development set:
0.913 (+/-0.041) for {'n_neighbors': 1}
0.887 (+/-0.053) for {'n_neighbors': 2}
0.907 (+/-0.070) for {'n_neighbors': 3}
0.887 (+/-0.067) for {'n_neighbors': 4}
0.899 (+/-0.073) for {'n_neighbors': 5}
Recall metric in the testing dataset:  0.9115646258503401
```

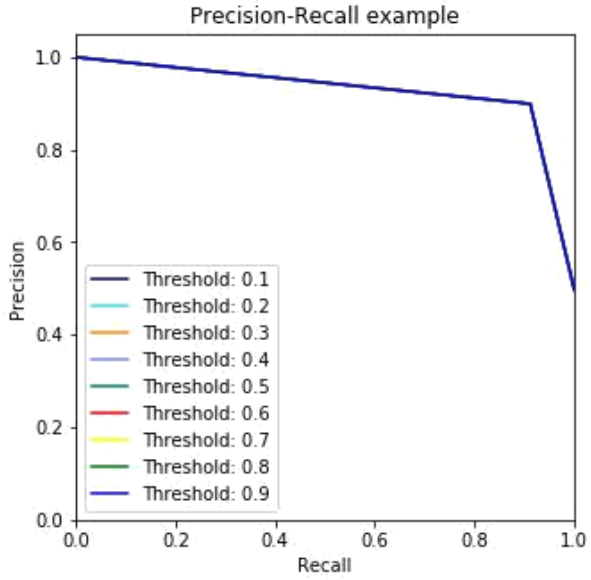



```
Recall metric in the testing dataset:  0.7891156462585034
```



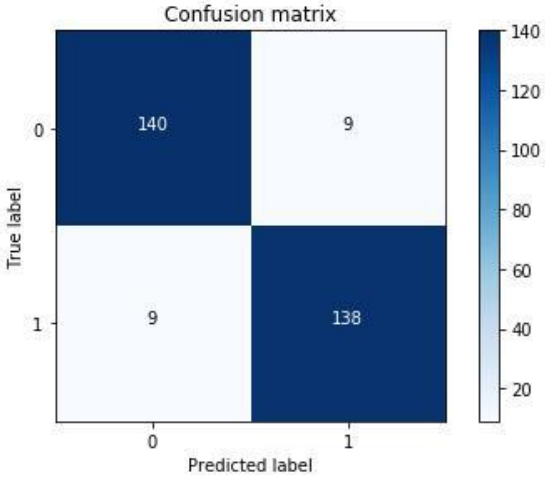

```
Recall metric in the whole testing dataset for threshold 0.1:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.2:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.3:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.4:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.5:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.6:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.7:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.8:
0.9115646258503401
Recall metric in the whole testing dataset for threshold 0.9:
0.9115646258503401
```



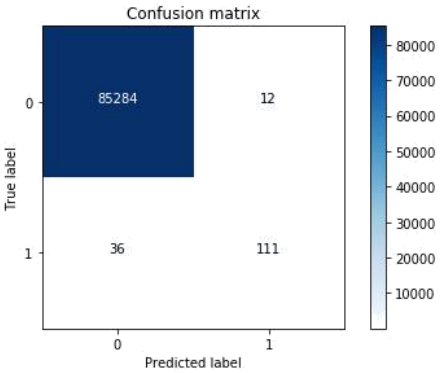


DecisionTree:

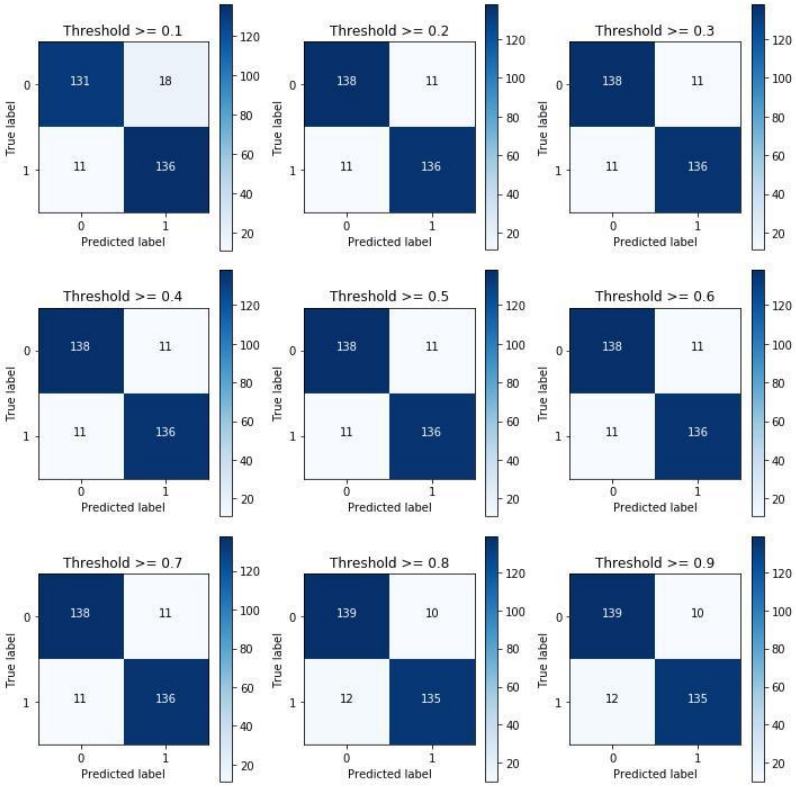
```
Best parameters set found on development set:
{'max_depth': 7}

Grid scores on development set:
0.904 (+/-0.085) for {'max_depth': 3}
0.907 (+/-0.070) for {'max_depth': 4}
0.904 (+/-0.089) for {'max_depth': 5}
0.907 (+/-0.054) for {'max_depth': 6}
0.913 (+/-0.064) for {'max_depth': 7}
Recall metric in the testing dataset:  0.9387755102040817
```

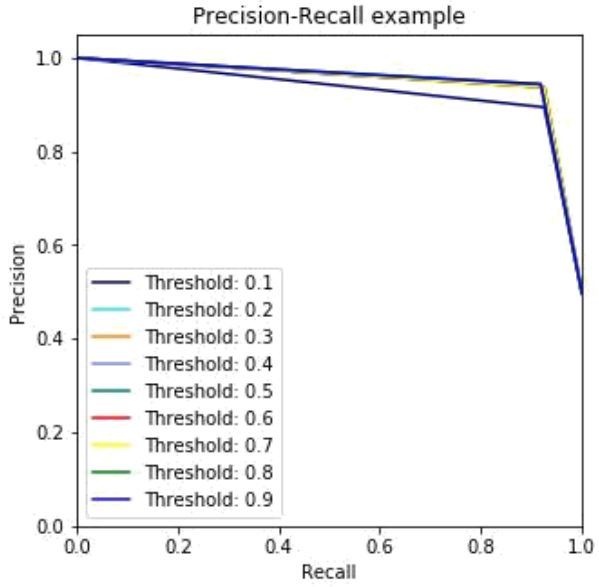



```
Recall metric in the testing dataset:  0.7551020408163265
```



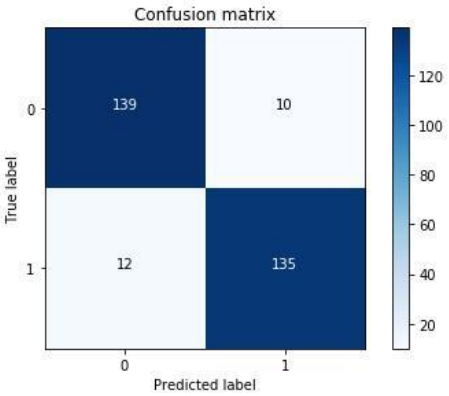
Recall metric in the whole testing dataset for threshold 0.1:
0.9251700680272109
Recall metric in the whole testing dataset for threshold 0.2:
0.9251700680272109
Recall metric in the whole testing dataset for threshold 0.3:
0.9251700680272109
Recall metric in the whole testing dataset for threshold 0.4:
0.9251700680272109
Recall metric in the whole testing dataset for threshold 0.5:
0.9251700680272109
Recall metric in the whole testing dataset for threshold 0.6:
0.9251700680272109
Recall metric in the whole testing dataset for threshold 0.7:
0.9251700680272109
Recall metric in the whole testing dataset for threshold 0.8:
0.9183673469387755
Recall metric in the whole testing dataset for threshold 0.9:
0.9183673469387755



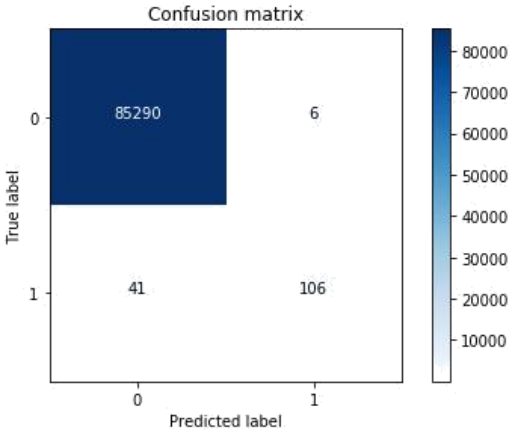


RamdomForest:

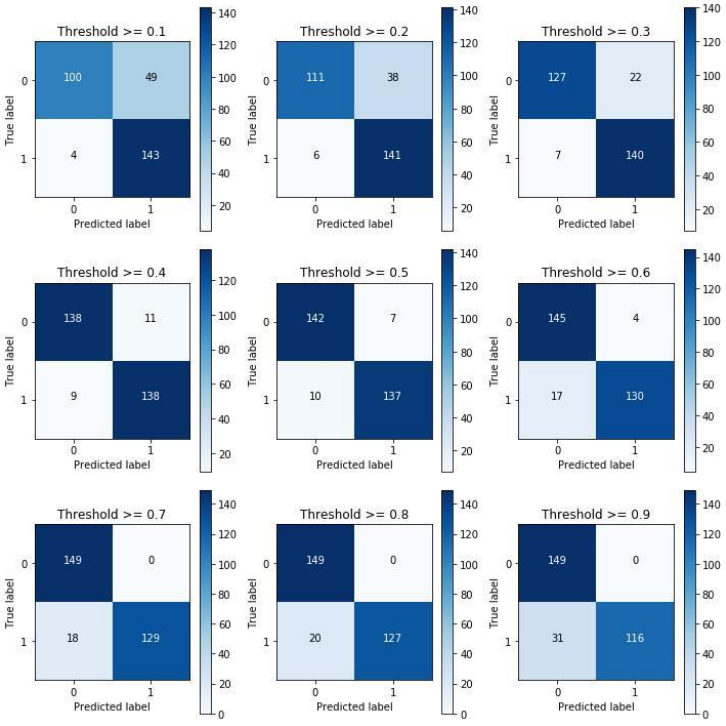
```
Best parameters set found on development set:
{'n_estimators': 20}

Grid scores on development set:
0.901 (+/-0.056) for {'n_estimators': 5}
0.890 (+/-0.062) for {'n_estimators': 10}
0.907 (+/-0.060) for {'n_estimators': 20}
0.907 (+/-0.072) for {'n_estimators': 50}
Recall metric in the testing dataset:  0.9183673469387755
```

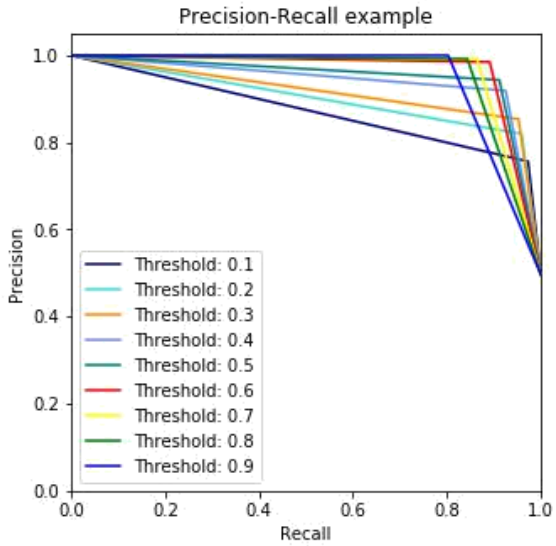



```
Recall metric in the testing dataset:  0.7210884353741497
```




```
Recall metric in the whole testing dataset for threshold 0.1:
0.9727891156462585
Recall metric in the whole testing dataset for threshold 0.2:
0.9591836734693877
Recall metric in the whole testing dataset for threshold 0.3:
0.9523809523809523
Recall metric in the whole testing dataset for threshold 0.4:
0.9387755102040817
Recall metric in the whole testing dataset for threshold 0.5:
0.9319727891156463
Recall metric in the whole testing dataset for threshold 0.6:
0.8843537414965986
Recall metric in the whole testing dataset for threshold 0.7:
0.8775510204081632
Recall metric in the whole testing dataset for threshold 0.8:
0.8639455782312925
Recall metric in the whole testing dataset for threshold 0.9:
0.7891156462585034
```





Submission:

Submit the following files:

1. Your program source code (e.g. updated task10_1.ipynb)
2. A screen shot of your program running Or

Your jupyter notebook (e.g. task10_1.ipynb)

Check the following things before submitting:

1. Add proper comments to your code