algorithms

March 14, 2021

1 Comparison of various algorithms on metrics

This notebook for running algorithms: * Logistic Regression * SVM * Random Forests * Artificial neural network

and scores them based on metrics: * Accuracy * F1 Score * ROC AUC * Precision * Recall

Runs each algorithm for each dataset across 5 trials, where GridSearch is used to find the optimal hyperparameters for each metric, then runs the classifier on training/testing sets and takes the mean over 5 trials.

Results are then stored in the results folder of this directory

[6]:
```
# import needed packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score,␣
 ↪precision_score, recall_score
from sklearn.preprocessing import StandardScaler

# import needed functions
from preprocess import prep_airlines, prep_income, prep_phishing, prep_surgical
from bootstrap import bootstrap
from logistic_regression import run_logistic_regression
from random_forest import run_random_forests
from support_vector import run_svm
from artificial_nn import run_ann

import warnings
warnings.filterwarnings('ignore')
```

1.1 Define datasets and metrics to be used

[2]:
```
# datasets and metrics
datasets = ['airline', 'income', 'phishing', 'surgical']
metrics = ['accuracy', 'f1', 'roc_auc', 'precision', 'recall']
```

2 Logistic Regression

[3]:
```
# final values
logreg_results_train = np.zeros((len(datasets), len(metrics)))
logreg_results_test = np.zeros((len(datasets), len(metrics)))
logreg_hyperparams = [] # list of dataframes

# for each dataset: run trials and add to final results

# AIRLINE
print('AIRLINE\n----------------------------------')
X,y = prep_airlines()
train, test, hypers = run_logistic_regression(X,y)

logreg_results_train[0,:] = train
logreg_results_test[0,:] = test
logreg_hyperparams.append(hypers)

# INCOMES
print('\nINCOMES\n----------------------------------')
X,y = prep_income()
train, test, hypers = run_logistic_regression(X,y)

logreg_results_train[1,:] = train
logreg_results_test[1,:] = test
logreg_hyperparams.append(hypers)

# PHISHING
print('\nPHISHING\n----------------------------------')
X,y = prep_phishing()
train, test, hypers = run_logistic_regression(X,y)

logreg_results_train[2,:] = train
logreg_results_test[2,:] = test
logreg_hyperparams.append(hypers)

# SURGICAL
print('\nSURGICAL\n----------------------------------')
X,y = prep_surgical()
train, test, hypers = run_logistic_regression(X,y)
```

```
logreg_results_train[3,:] = train
logreg_results_test[3,:] = test
logreg_hyperparams.append(hypers)
```

```
AIRLINE
----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

INCOMES
----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

PHISHING
----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

SURGICAL
----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done
```

[23]:
```
# save to results folder
result_dir = './results/'

df_results_train = pd.DataFrame(logreg_results_train, columns=metrics,␣
 ↪index=datasets)
df_results_train['mean'] = np.mean(df_results_train, axis=1)
df_results_train.to_csv(result_dir+'logreg_results_train.csv')

df_results_test = pd.DataFrame(logreg_results_test, columns=metrics,␣
 ↪index=datasets)
```

```
df_results_test['mean'] = np.mean(df_results_test, axis=1)
df_results_test.to_csv(result_dir+'logreg_results_test.csv')

df_hyperparams = pd.concat(logreg_hyperparams)
df_hyperparams.to_csv(result_dir+'logreg_hyperparameters.csv')
```

[ ]:
```
# to visualize hyperparameter search results
for i,hyp in enumerate(logreg_hyperparams):
    sns.heatmap(hyp, annot=True, cmap='viridis')
    plt.title(datasets[i])
    plt.show()
```

3 SVM

[3]:
```
# final values
svm_results_train = np.zeros((len(datasets), len(metrics)))
svm_results_test = np.zeros((len(datasets), len(metrics)))

# for each dataset: run trials and add to final results

# AIRLINE
print('AIRLINE\n-----------------------------------')
X,y = prep_airlines()
train, test = run_svm(X,y)

svm_results_train[0,:] = train
svm_results_test[0,:] = test

# INCOMES
print('\nINCOMES\n-----------------------------------')
X,y = prep_income()
train, test = run_svm(X,y)

svm_results_train[1,:] = train
svm_results_test[1,:] = test

# PHISHING
print('\nPHISHING\n-----------------------------------')
X,y = prep_phishing()
train, test = run_svm(X,y)

svm_results_train[2,:] = train
svm_results_test[2,:] = test

# SURGICAL
print('\nSURGICAL\n-----------------------------------')
```

```
X,y = prep_surgical()
train, test = run_svm(X,y)

svm_results_train[3,:] = train
svm_results_test[3,:] = test
```

```
AIRLINE
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

INCOMES
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

PHISHING
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

SURGICAL
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done
```

[4]:
```
# save to results folder
result_dir = './results/'

df_results_train = pd.DataFrame(svm_results_train, columns=metrics,␣
 ↪index=datasets)
df_results_train['mean'] = np.mean(df_results_train, axis=1)
df_results_train.to_csv(result_dir+'svm_results_train.csv')
```

```
df_results_test = pd.DataFrame(svm_results_test, columns=metrics,␣
↪index=datasets)
df_results_test['mean'] = np.mean(df_results_test, axis=1)
df_results_test.to_csv(result_dir+'svm_results_test.csv')
```

4 Random Forests

[3]:
```
# final values
rf_results_train = np.zeros((len(datasets), len(metrics)))
rf_results_test = np.zeros((len(datasets), len(metrics)))
rf_hyperparams = [] # list of dataframes

# for each dataset: run trials and add to final results

# AIRLINE
print('AIRLINE\n-----------------------------------')
X,y = prep_airlines()
train, test, hypers = run_random_forests(X,y)

rf_results_train[0,:] = train
rf_results_test[0,:] = test
rf_hyperparams.append(hypers)

# INCOMES
print('\nINCOMES\n-----------------------------------')
X,y = prep_income()
train, test, hypers = run_random_forests(X,y)

rf_results_train[1,:] = train
rf_results_test[1,:] = test
rf_hyperparams.append(hypers)

# PHISHING
print('\nPHISHING\n-----------------------------------')
X,y = prep_phishing()
train, test, hypers = run_random_forests(X,y)

rf_results_train[2,:] = train
rf_results_test[2,:] = test
rf_hyperparams.append(hypers)

# SURGICAL
print('\nSURGICAL\n-----------------------------------')
X,y = prep_surgical()
train, test, hypers = run_random_forests(X,y)
```

```
rf_results_train[3,:] = train
rf_results_test[3,:] = test
rf_hyperparams.append(hypers)
```

```
AIRLINE
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

INCOMES
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

PHISHING
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

SURGICAL
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done
```

[4]:
```
# save to results folder
result_dir = './results/'

df_results_train = pd.DataFrame(rf_results_train, columns=metrics,␣
 ↪index=datasets)
df_results_train['mean'] = np.mean(df_results_train, axis=1)
df_results_train.to_csv(result_dir+'rf_results_train.csv')

df_results_test = pd.DataFrame(rf_results_test, columns=metrics, index=datasets)
df_results_test['mean'] = np.mean(df_results_test, axis=1)
df_results_test.to_csv(result_dir+'rf_results_test.csv')
```

```
df_hyperparams = pd.concat(rf_hyperparams)
df_hyperparams.to_csv(result_dir+'rf_hyperparameters.csv')
```

5 Artifical Neural Networks

[7]:
```
# final values
ann_results_train = np.zeros((len(datasets), len(metrics)))
ann_results_test = np.zeros((len(datasets), len(metrics)))
ann_hyperparams = [] # list of dataframes

# for each dataset: run trials and add to final results

# AIRLINE
print('AIRLINE\n---------------------------------')
X,y = prep_airlines()
train, test, hypers = run_ann(X,y)

ann_results_train[0,:] = train
ann_results_test[0,:] = test
ann_hyperparams.append(hypers)

# INCOMES
print('\nINCOMES\n---------------------------------')
X,y = prep_income()
# doesn't work with non-scaled values of second feature
scaler = StandardScaler()
X[:,1] = scaler.fit_transform(X[:,1].reshape(-1,1)).reshape(-1)
train, test, hypers = run_ann(X,y)

ann_results_train[1,:] = train
ann_results_test[1,:] = test
ann_hyperparams.append(hypers)

# PHISHING
print('\nPHISHING\n---------------------------------')
X,y = prep_phishing()
train, test, hypers = run_ann(X,y)

ann_results_train[2,:] = train
ann_results_test[2,:] = test
ann_hyperparams.append(hypers)

# SURGICAL
print('\nSURGICAL\n---------------------------------')
X,y = prep_surgical()
```

```
train, test, hypers = run_ann(X,y)

ann_results_train[3,:] = train
ann_results_test[3,:] = test
ann_hyperparams.append(hypers)
```

```
AIRLINE
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

INCOMES
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

PHISHING
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done

SURGICAL
-----------------------------------
Trial 1 done
Trial 2 done
Trial 3 done
Trial 4 done
Trial 5 done
```

[8]:
```
# save to results folder
result_dir = './results/'

df_results_train = pd.DataFrame(ann_results_train, columns=metrics,␣
 ↪index=datasets)
df_results_train['mean'] = np.mean(df_results_train, axis=1)
df_results_train.to_csv(result_dir+'ann_results_train.csv')
```

```
df_results_test = pd.DataFrame(ann_results_test, columns=metrics,␣
↪index=datasets)
df_results_test['mean'] = np.mean(df_results_test, axis=1)
df_results_test.to_csv(result_dir+'ann_results_test.csv')

df_hyperparams = pd.concat(ann_hyperparams)
df_hyperparams.to_csv(result_dir+'ann_hyperparameters.csv')
```

[ ]: