bnml

September 29, 2020

```
[1]: #read in data
     import pandas as pd
     data = pd.read_csv('/Users/nicksbox/Documents/Data/BN/bnmldata.csv')
```

```
[3]: data.shape
```

```
[3]: (244857, 8)
```

0.0.1 ML Train & Test

```
[4]: #label target
     cols =[col for col in data.columns if col not in ['Label']]
     mldata = data[cols]
     target = data['Label']
```

```
[11]: from sklearn.preprocessing import StandardScaler
      from sklearn.pipeline import make_pipeline
      from sklearn.model_selection import train_test_split
      from sklearn.metrics import accuracy_score
      from sklearn.metrics import recall_score
      from sklearn.metrics import f1_score
      from sklearn.metrics import roc_auc_score
      from sklearn.metrics import roc_curve, auc
      from sklearn.metrics import average_precision_score
      import matplotlib.pyplot as plt
      import joblib
      from joblib import dump, load
```

```
[6]: #train test split
     X = mldata
     y = target
     X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.3,␣
      ↪random_state=0)
```

0.0.2 SVM Classifier

[ ]:

```
#SVM
from sklearn import svm
clf = svm.SVC()
y_pred = clf.fit(X_train,y_train).predict(X_test)
```

[23]:

```
#model performance
accuracy_score = accuracy_score(y_test, y_pred)
print("Accuracy Score:", accuracy_score)
average_precision = average_precision_score(y_test, y_pred)
print("Precision:", average_precision)
recall = recall_score(y_test, y_pred)
print("Recall:", recall)
f1 = f1_score(y_test, y_pred)
print("F1 Score:", f1)
AUROC = roc_auc_score(y_test, y_pred)
print("AUROC:", AUROC)
```

```
Accuracy Score: 0.7327588554003648
Precision: 0.7327588554003648
Recall: 1.0
F1 Score: 0.8457713006245826
AUROC: 0.5
```

[18]:

```
#serialize SVM
import joblib
from joblib import dump, load
bn_pisvm = 'bn_svm.sav'
joblib.dump(clf, bn_pisvm)
```

[18]:

```
['bn_svm.sav']
```

0.0.3 Naive Bayes Classifier

[7]:

```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
y_pred = gnb.fit(X_train,y_train).predict(X_test)
```

[8]:

```
#model performance
#accuracy_score = accuracy_score(y_test, y_pred)
#print("Accuracy Score:", accuracy_score)
average_precision = average_precision_score(y_test, y_pred)
print("Precision:", average_precision)
recall = recall_score(y_test, y_pred)
print("Recall:", recall)
f1 = f1_score(y_test, y_pred)
```

```
print("F1 Score:", f1)
AUROC = roc_auc_score(y_test, y_pred)
print("AUROC:", AUROC)
```

```
Precision: 0.7579879740329235
Recall: 1.0
F1 Score: 0.8623357898109579
AUROC: 0.562273954459783
```

[12]:

```
#serialize NBC
from joblib import dump, load
bn_pinbc = 'bn_nbc.sav'
joblib.dump(gnb, bn_pinbc)
```

[12]:

```
['bn_nbc.sav']
```

0.0.4 Logistic Regression Classifier

[15]:

```
from sklearn.linear_model import LogisticRegression
lrclf = LogisticRegression(random_state=0)
y_pred = lrclf.fit(X_train,y_train).predict(X_test)
```

[16]:

```
#model performance
#accuracy_score = accuracy_score(y_test, y_pred)
#print("Accuracy Score:", accuracy_score)
average_precision = average_precision_score(y_test, y_pred)
print("Precision:", average_precision)
recall = recall_score(y_test, y_pred)
print("Recall:", recall)
f1 = f1_score(y_test, y_pred)
print("F1 Score:", f1)
AUROC = roc_auc_score(y_test, y_pred)
print("AUROC:", AUROC)
```

```
Precision: 0.9508874691452125
Recall: 0.8603117394616084
F1 Score: 0.9190101014110222
AUROC: 0.9137787111550821
```

[17]:

```
#serialize LogReg
from joblib import dump, load
bn_pilogreg = 'bn_logreg.sav'
joblib.dump(lrclf, bn_pilogreg)
```

[17]:

```
['bn_logreg.sav']
```

[ ]:

```

```