bnvalidationml

September 29, 2020

0.1 Predictions Using Validation Dataset

[60]:
```
import pandas as pd
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
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
import matplotlib.pyplot as plt
from joblib import dump, load
```

[61]:
```
#load previously trained, persistent models
svmclf = load('bn_svm.sav')
nbcclf = load('bn_nbc.sav')
logregclf = load('bn_logreg.sav')
```

[62]:
```
#read in validation patient dataset
valptdf = pd.read_csv('/Users/nicksbox/Documents/Data/BN/validationpts.csv')
```

[63]:
```
valptdf = valptdf.drop(['Unnamed: 0'], axis=1)
```

[66]:
```
valptdf.shape
```

[66]:
```
(11183, 8)
```

[67]:
```
data = valptdf
```

0.1.1 Make Predictions on Validation Dataset with Models

[68]:
```
#test models
#label target
cols =[col for col in data.columns if col not in ['Label']]
X = data[cols]
y = data['Label']
```

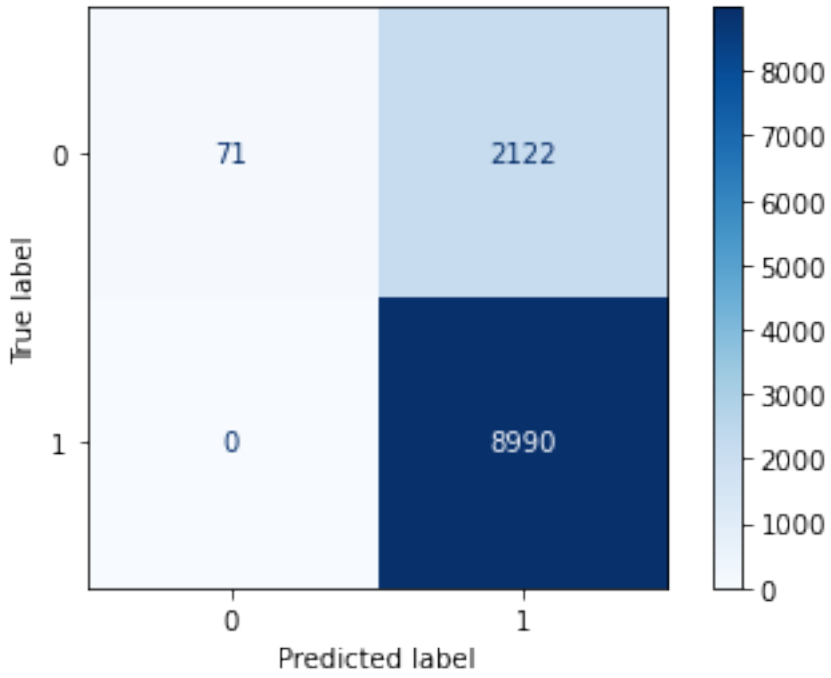
Logistic Regression Classifier Predictions

[69]:
```
#test LogReg Classifier
yhat = logregclf.predict(X)
```

[70]:
```
#confusion matrix for Logreg
confusion_matrix(y, yhat)
```

[70]:
```
array([[  71, 2122],
       [   0, 8990]])
```

[85]:
```
plot_confusion_matrix(logregclf, X,y, cmap=plt.cm.Blues)
plt.show()
```



[71]:
```
#model performance
average_precision = average_precision_score(yhat, y)
print("Precision:", average_precision)
```

```
recall = recall_score(yhat, y)
print("Recall:", recall)
f1 = f1_score(yhat, y)
print("F1 Score:", f1)
AUROC = roc_auc_score(yhat, y)
print("AUROC:", AUROC)
```

```
Precision: 0.9987875797799898
Recall: 0.8090352771778258
F1 Score: 0.8944383643418565
AUROC: 0.9045176385889129
```

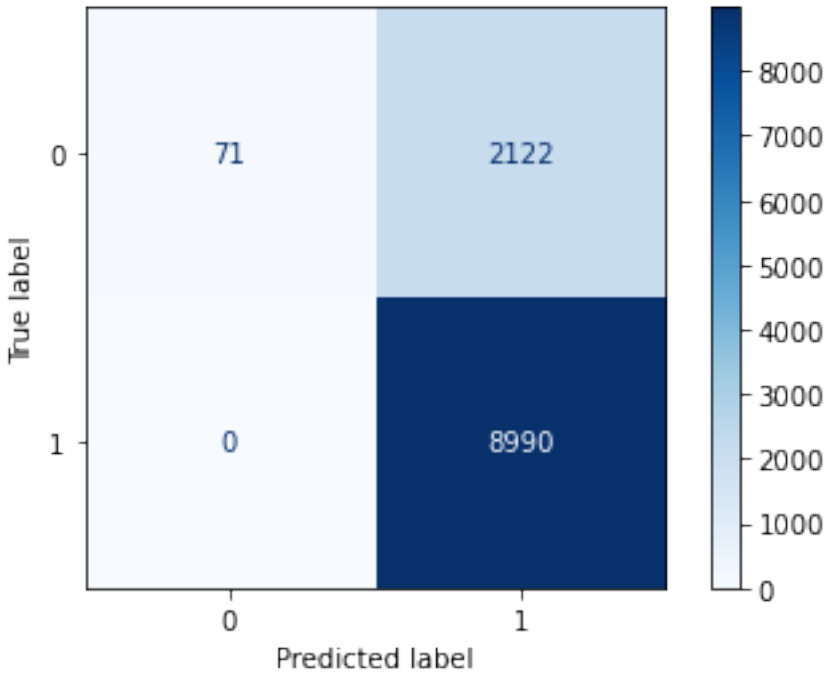
SVM Classifier Predictions

[72]:
```
#test SVM Classifier
yhat = svmclf.predict(X)
```

[73]:
```
#confusion matrix for SVM
confusion_matrix(y, yhat)
```

[73]:
```
array([[  71, 2122],
       [   0, 8990]])
```

[83]:
```
plot_confusion_matrix(svmclf, X,y, cmap=plt.cm.Blues)
plt.show()
```



[74]:
```
#model performance
average_precision = average_precision_score(yhat, y)
print("Precision:", average_precision)
recall = recall_score(yhat, y)
print("Recall:", recall)
f1 = f1_score(yhat, y)
print("F1 Score:", f1)
AUROC = roc_auc_score(yhat, y)
print("AUROC:", AUROC)
```

```
Precision: 0.9987875797799898
Recall: 0.8090352771778258
F1 Score: 0.8944383643418565
AUROC: 0.9045176385889129
```

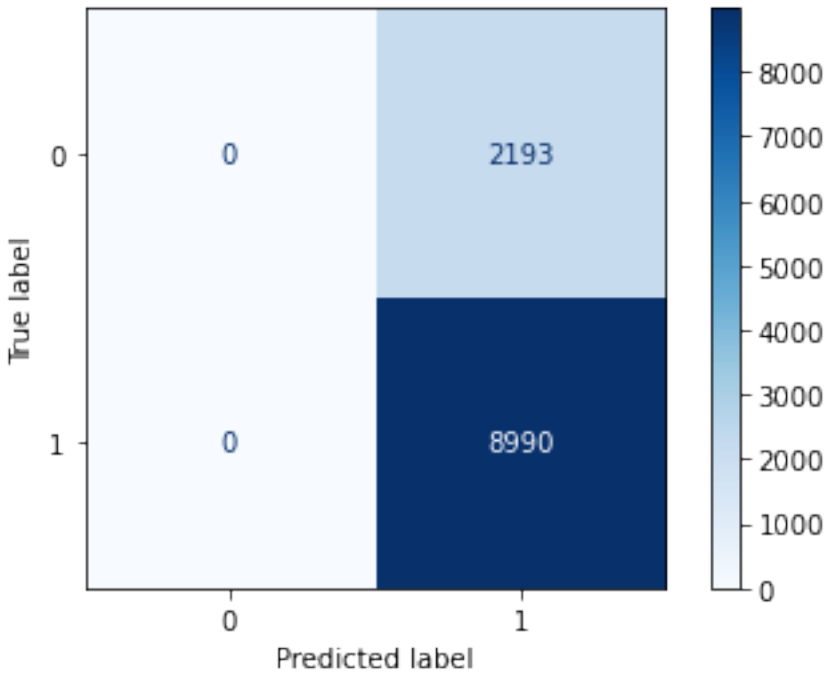
Naive Bayes Classifier Predictions

[75]:
```
#BNC
yhat = nbcclf.predict(X)
```

[76]:
```
#confusion matrix for NBC
confusion_matrix(y, yhat)
```

[76]:
```
array([[   0, 2193],
       [   0, 8990]])
```

[84]:
```
plot_confusion_matrix(nbcclf, X,y, cmap=plt.cm.Blues)
plt.show()
```



[88]:
```
#model performance
average_precision = average_precision_score(yhat, y)
print("Precision:", average_precision)
recall = recall_score(yhat, y)
print("Recall:", recall)
f1 = f1_score(yhat, y)
print("F1 Score:", f1)
```

```
Precision: 1.0
Recall: 0.8038987749262273
F1 Score: 0.8912903385713576
```

[ ]: