Chunyuan Shen

ID:801322013

Homework 2

https://github.com/cryyin/ECGR5105/tree/main/homework3

Problem 1 (40pts):

Use the cancer dataset to build a Naïve Bayesian model to classify the type of cancer (Malignant vs. benign). Plot your classification accuracy, precision, and recall. Explain and elaborate on your results. Can you compare your results against the logistic regression classifier you did in previous homework.

Naïve Bayesian model：

```
from sklearn.naive_bayes import GaussianNB
```

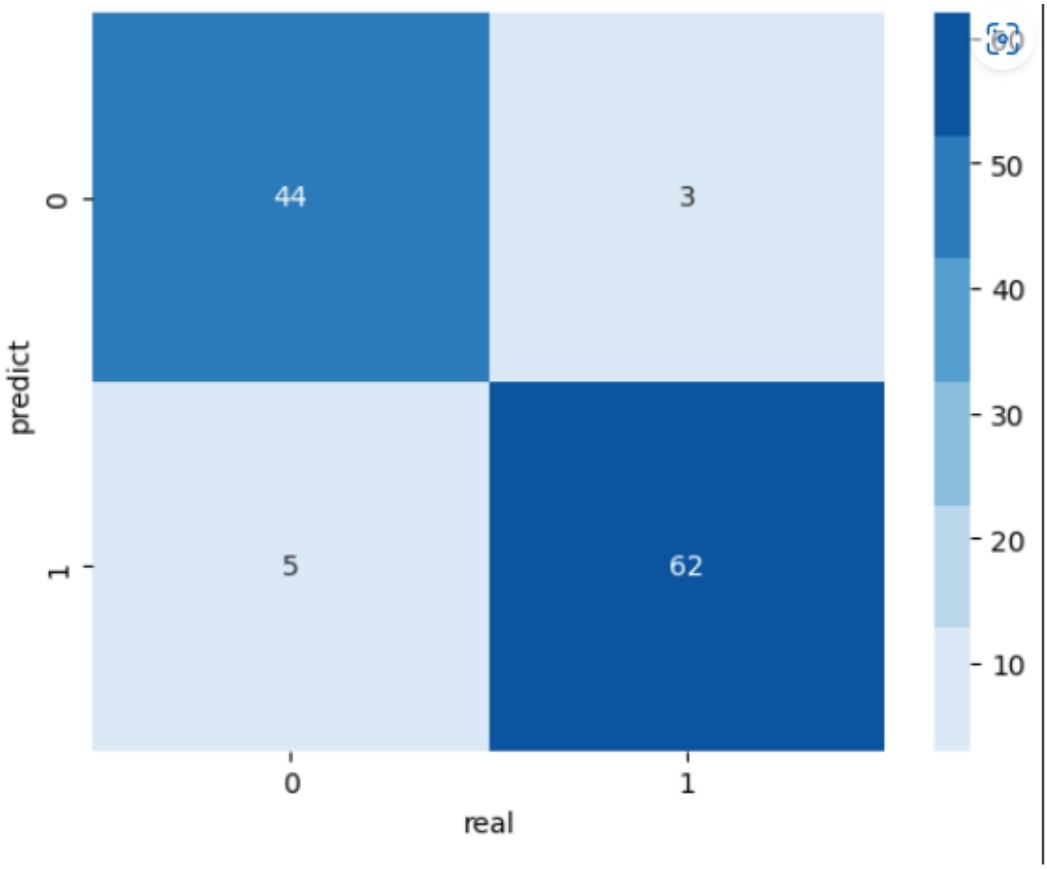
```
clf=GaussianNB()
clf.fit(x_train,y_train)#对训练集进行拟合
Y_pred= clf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, Y_pred))
print("Precision:",metrics.precision_score(y_test, Y_pred))
print("Recall:",metrics.recall_score(y_test, Y_pred))
```

We know that Bayes' theorem is:$P(A \mid B) = \frac{P(B|A)P(A)}{P(B)}$

Use machine learning x, y to replace events A and B to get the formula could be used in machine learning.

Result：

Accuracy: 0.9298245614035088
Precision: 0.9253731343283582
Recall: 0.9538461538461539



In the previous homework its result is 0.9473684210526315, better than this time's.

Problem 2 (40pts):

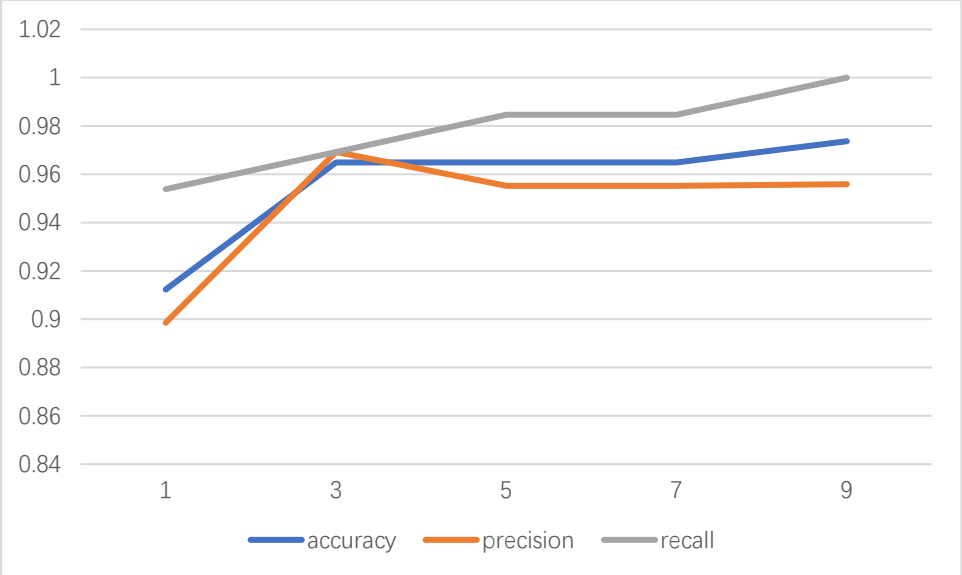
Use the cancer dataset to build a logistic regression model to classify the type of cancer (Malignant vs. benign). Use the PCA feature extraction for your training. Perform N number of independent training (N=1, ..., K). Identify the optimum number of K, principal components that achieve the highest classification accuracy. Plot your classification accuracy, precision, and recall over a different number of Ks. Explain and elaborate on your results.

When k=9 it has the best performance:

Accuracy: 0.9736842105263158
Precision: 0.9558823529411765
Recall: 1.0

k	accuracy	precision	recall
1	0.912281	0.8985507	0.9538462
3	0.9649123	0.9692308	0.9692308
5	0.9649123	0.9552239	0.9846154

7	0.9649123	0.9552239	0.9846154
9	0.9736842	0.9558824	1



But when k=3 it has the best precision, that might because when we going to lower dimensions we will loss some information.

Problem 3 (20pts):

Can you repeat problem 2? This time, replace logistic regression with the Bayes classifier. Report your results (classification accuracy, precision, and recall). Compare your results against problem 2.

```
k=1    #设置降维的占比
pca= PCA(n_components=k)#调用PCA函数，先实例化
pcaCom = pca.fit_transform(x)
pcaCom = pd.DataFrame(pcaCom)
print("主成分的数量：",pca.n_components_)
X = pcaCom.iloc[:, [0]].values
#X = pcaCom.iloc[:, [0, 1, 2, 3, 4, 5, 6, 7, 8]].values
Y = breast_dataset.iloc[:, 30].values
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, trair
```

```
主成分的数量： 1
```

```
model = GaussianNB()
model.fit(X_train, Y_train)
Y_pred= model.predict(X_test)
Y_pred[0:9]
```

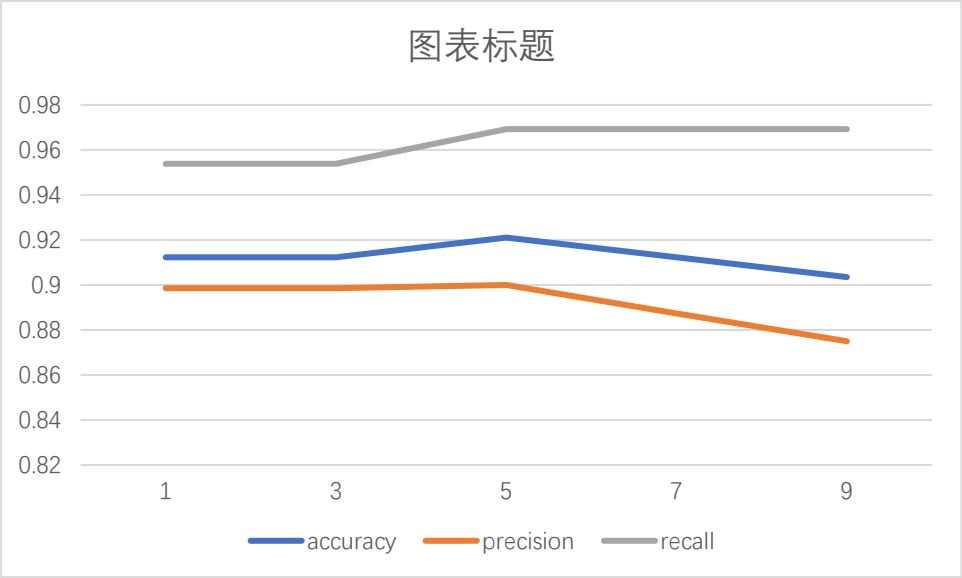
When k=5 it has the best performance:

Accuracy: 0.9210526315789473

Precision: 0.9

Recall: 0.9692307692307692

k	accuracy	precision	recall
1	0.9122807	0.8985507	0.9538462
3	0.9122807	0.8985507	0.9538462
5	0.9210526	0.9	0.9692308
7	0.9122807	0.8873239	0.9692308
9	0.9035088	0.875	0.9692308



The results are not as good as previous one, but it reach the best performance at 5.