نام و نام خانوادگی: حسنا اویارحسینی	شماره دانشجویی: ۹۸۲۳۳۰۸
نام درس: مقدماتی بر یادگیری ماشین	شماره تمرین: تمرین شماره ۳

((گزارش تمرین علمی بیزین))

برای پیاده سازی هر دو بخش الف و ب از دو تابع زیر استفاده میکنیم در تابع اول مدل و fold های مختلف که از روی داده های آموزشی ساختیم و داده های آموزشی را به عنوان ورودی میدهیم. Fold ها به کمک تابع fold در کتابخانه scikit learn ساخته میشوند این تابع شاخص‌های آموزش/تست را برای تقسیم داده‌ها در مجموعه‌های آموزش/تست فراهم می‌کند. مجموعه داده را به k تاهای متوالی تقسیم میکند. سپس هر فولد یک بار به عنوان اعتبار سنجی استفاده می شود در حالی که k - 1 فولد باقی مانده مجموعه آموزشی را تشکیل می دهد.

این تابع در یک حلقه هر بار یک فولد را به عنوان داده تست میگیرد و با بقیه فولد ها مدل را آموزش میدهد و نتایج حاصل از پیش بینی بر روی داده های تست را در آرایه predicted_class ذخیره میکند. در آخر نیز دقت را با توجه به برچسب های اولیه برای این داده محاسبه و ذخیره میکند. در نهایت این تابع کلاس های صحیح و پیش بینی شده و دقت میانگین را برمیگرداند.

تابع دوم برای رسم confusin matrix می باشد که از روی کلاس های صحیح و پیش بینی شده این ماتریس را ساخته و نمایش میدهد.

predicts and confusion matrix functions

[6]:
```python
def cross_val_predict(model, kfold : KFold, X : np.array, y : np.array):

    model_ = cp.deepcopy(model)

    actual_classes = np.empty([0], dtype=int)
    predicted_classes = np.empty([0], dtype=int)
    accuracy = np.empty([0], dtype=int)

    for train_ndx, test_ndx in kfold.split(X):
        # Extracts the rows from the data for the training and testing
        train_X, train_y, test_X, test_y = X[train_ndx], y[train_ndx], X[test_ndx], y[test_ndx]
        # Appends the actual target classifications to actual_classes
        actual_classes = np.append(actual_classes, test_y)
        # Fits the machine learning model using the training data extracted from the current fold
        model_.fit(train_X, train_y)
        # Uses the fitted model to predict the target classifications for the test data in the current fold
        predicted_classes = np.append(predicted_classes, model_.predict(test_X))
        accuracy = np.append(accuracy, accuracy_score(actual_classes, predicted_classes))

    return actual_classes, predicted_classes, accuracy.mean()
```

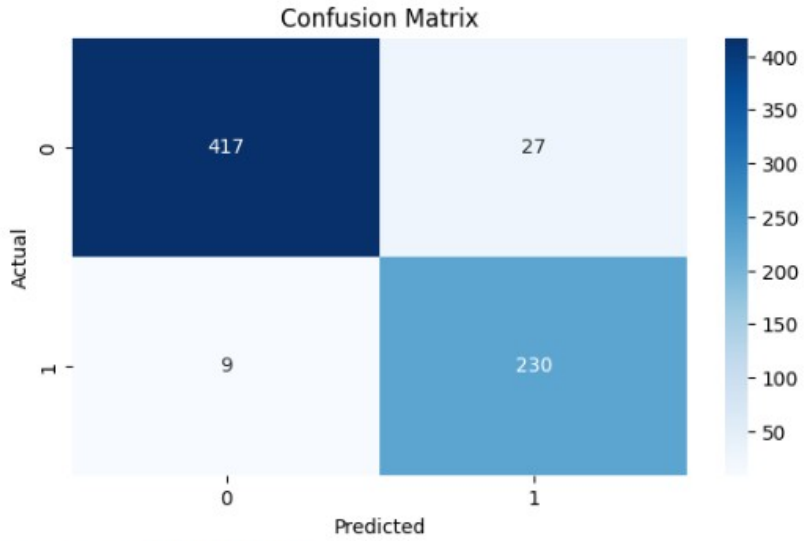
[7]:
```python
def plot_confusion_matrix(actual_classes : np.array, predicted_classes : np.array, sorted_labels : list):

    matrix = confusion_matrix(actual_classes, predicted_classes, labels=sorted_labels)

    plt.figure(figsize=(7,4))
    sns.heatmap(matrix, annot=True, xticklabels=sorted_labels, yticklabels=sorted_labels, cmap="Blues", fmt="g")
    plt.xlabel('Predicted'); plt.ylabel('Actual'); plt.title('Confusion Matrix')

    plt.show()
```

الف)

در این قسمت از مدل naive bayes از نوع multinomial استفاده شده است که این تابع در کتابخانه scikit learn وجود دارد سپس با تقسیم داده ها به ۵ فولد مدل به کمک توابع توضیح دده شده آموزشداده شده و نتایج بر روی داده های تست در قالب ماتریس آشفتگی نمایش داده شده است.

```
[8]: model = MultinomialNB()
     kfold = KFold(n_splits=5, random_state=42, shuffle=True)
```

```
[9]: actual_classes, predicted_classes, accuracy = cross_val_predict(model, kfold, X.to_numpy(), y.to_numpy())
     plot_confusion_matrix(actual_classes, predicted_classes, [0, 1])
     print(f"accuracy: {accuracy}")
```




```
accuracy: 0.9534298581066205
```

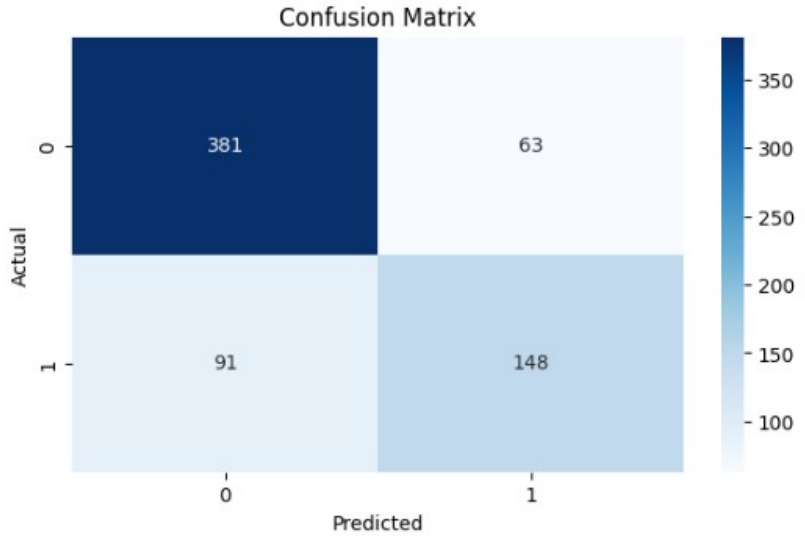
ب)

این بخش مانند قسمت قبل است با این تفاوت که اینجا از مدل Gaussian naive bayes استفاده کردیم که در کتابخانه scikit learn وجود دارد.
در ادامه نتایج را مقایسه میکنیم.

b)

[10]:
```
model = GaussianNB()
kfold = KFold(n_splits=5, random_state=42, shuffle=True)
```

[11]:
```
actual_classes, predicted_classes, accuracy = cross_val_predict(model, kfold, X.to_numpy(), y.to_numpy())
plot_confusion_matrix(actual_classes, predicted_classes, [0, 1])
print(f"accuracy: {accuracy}")
```




```
accuracy: 0.7523783322946164
```

همان طور که مشاهده میکنیم عملکرد و دقت مدل اول یعنی multinomial بهتر بوده است دقت حدود ۲۰٪ بهتر بوده است این موضوع نشان میدهد که توزیع داده های ما احتمالا بیشتر شبیه با multinomial بوده است تا gussian و به همین علت عملکرد بهتری در بخش الف داشته ایم.