

جواب سوال 4)

```

from sklearn.utils.validation import check_X_y, check_array, check_is_fitted
from sklearn.model_selection import train_test_split as tts
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

# perform Gaussian naive_bayes without package
x1 = ds[['trestbps', 'chol', 'thalach']]
X_train, X_test, y_train, y_test = tts(x1, y, test_size = 0.2, random_state = 0)
class GaussianNB1(BaseEstimator, ClassifierMixin):
    def fit(self, X, y):
        X, y = check_X_y(X, y)
        self.prior = np.bincount(y) / len(y)
        self.max1 = np.max(y) + 1

        self.mean1 = np.array([X[np.where(y==i)].mean(axis=0) for i in range(self.max1)])
        self.std1 = np.array([X[np.where(y==i)].std(axis=0) for i in range(self.max1)])
        return self
    def predict_prob(self, X):
        check_is_fitted(self)
        X = check_array(X)

        res = []
        for i in range(len(X)):
            prob = []
            for j in range(self.max1):
                prob.append((1/np.sqrt(2*np.pi*self.std1[j]**2))*np.exp(-0.5*((X[i]-
                    self.mean1[j])/self.std1[j])**2)).prod()*self.priors_[j])
            prob = np.array(prob)
            res.append(prob / prob.sum())
        return np.array(res)

    def predict(self, X):
        check_is_fitted(self)
        X = check_array(X)
        res = self.predict_prob(X)
        return res.argmax(axis=1)

gnb = GaussianNB1()
gnb.fit(X_train, y_train)

```

می‌خواهیم مدل بی‌ز ساده گوسی را بدون استفاده از پکیج GaussianNB بسازیم. برای این منظور کلاس GaussianNB1 را ایجاد می‌کنیم و در آن سه تابع (fit و predict_prob و predict) را تعریف می‌کنیم. در تابع fit نمودار گوسی مدل را با استفاده از میانگین و انحراف معیار به دست می‌آوریم. در تابع predict_prob احتمال وجود داشتن داده را درون مدل گوسی بررسی می‌کنیم و در تابع predict با

استفاده از argmax مقدار مدل را روی محور x هنگامی که نمودار گوسی بیشینه است به دست می آوریم. یعنی مقدار را زمانی که به کلاس
تعلق دارد حساب می کنیم.