| 1 | et.head() nancies Glucose Blood 6 148 | Pressure Skin1 | Thickness Insul | n BMI Diabet | | ge Outcome | |
|---|--|--|--|--|--|--|--------------|
| 2 3 4 | 6 148 1 85 8 183 1 89 0 137 | 72 66 64 66 40 | 29 0 23 | 0 33.6 0 26.6 0 23.3 14 28.1 18 43.1 | 0.351 3 0.672 3 0.167 2 | 100 1 11 0 12 1 11 0 13 1 | |
| | et.head(20) gnancies Glucose Bloo 6 148 | dPressure Skir | Thickness Insu | lin BMI Diabe | tesPedigreeFunction A | Outcome 50 1 | _ |
| 1 2 3 4 | 1 85 8 183 1 89 0 137 | 66 64 66 40 | 29 0 23 35 | 0 26.60 23.394 28.168 43.1 | 0.351 0.672 0.167 2.288 | 31 0 32 1 21 0 33 1 | |
| 5 6 7 8 | 5 116 3 78 10 115 2 197 | 74 50 0 70 | 0 32 0 | 0 25.6 88 31.0 0 35.3 | 0.201 0.248 0.134 0.158 | 30 0 26 1 29 0 53 1 | |
| 9 10 11 | 8 125 4 110 10 168 | 96 92 74 | 0 0 0 | 0 0.0 0 37.6 0 38.0 | 0.232 0.191 0.537 | 54 1 30 0 34 1 | |
| 12 13 14 15 | 10 139 1 189 5 166 7 100 | 80 60 72 0 | | 0 27.1 346 30.1 75 25.8 0 30.0 | 1.441 0.398 0.587 0.484 | 57 0 59 1 51 1 32 1 | |
| 16 17 18 19 | 0 118 7 107 1 103 1 115 | 84 74 30 70 | 47 0 0 38 30 | 230 45.8 0 29.6 83 43.3 96 34.6 | 0.551 0.254 0.183 0.529 | 31 1 31 1 33 0 32 1 | |
| X = d | re considering all ataset.iloc[:, 0:8] ataset.iloc[:,8].va | .values | ur dataset a | s variables | | | |
| | [1, 0, 1, 0, 1, 0, | 1, 0, 1, 1] | , dtype=int | 54) | | | |
| array(| [[6.000e+00, 1.480e] 6.270e-01, 5.000e] [1.000e+00, 8.500e] 3.510e-01, 3.100e] [8.000e+00, 1.830e] 6.720e-01, 3.200e] | e+01], e+01, 6.600e e+01], e+02, 6.400e e+01], | +01, 2.900e- | -01, 0.000e+0 | 00, 2.660e+01, 00, 2.330e+01, | | |
| | [1.000e+00, 8.900e 1.670e-01, 2.100e [0.000e+00, 1.370e 2.288e+00, 3.300e [5.000e+00, 1.160e 2.010e-01, 3.000e [3.000e+00, 7.800e | e+01], e+02, 4.000e e+01], e+02, 7.400e e+01], e+01, 5.000e | +01, 3.500e- | -01, 1.680e+0 | 02, 4.310e+01, 00, 2.560e+01, | | |
| | 2.480e-01, 2.600 [1.000e+01, 1.150] 1.340e-01, 2.900] [2.000e+00, 1.970] 1.580e-01, 5.300] [8.000e+00, 1.250] 2.320e-01, 5.400] | e+01], e+02, 0.000e e+01], e+02, 7.000e e+01], e+02, 9.600e | +00, 0.000e- +01, 4.500e- | -00, 0.000e+0 | 00, 3.530e+01, 02, 3.050e+01, | | |
| from | ts dataset into Tessklearn.model_selection, X_test, Y_train | st and train ction import | train_test_ | | est_size = 0.20, | random_stat | e=0) |
| from sc_X : X_tra | ure scaling data sklearn.preprocess: = StandardScaler() in = sc_X.fit_trans t = sc_X.transform t[1] | sform(X_trai | | r | | | |
| – array(| [-0.54480808, -0.4 0.17619533, -0.1 it data into featur | 376381 , -0. | 88240283]) |).58457246, | 0.15216202, | | |
| testD | Data = np.asarray(Xata = np.asarray(Xata.shape | _ | | | | | |
| <pre>from class</pre> | rts Logistic regressives sklearn.linear_mode ifier = LogisticRegifier.fit(X_train, | el import Lo gression(ran | gisticRegres | | | | |
| Logis | icRegression(rando | om_state=0) | | | | | |
| Y_pre | <pre>icRegression(randor d = classifier.pred d[0:9]</pre> | | | | | | |
| array(| <pre>[1, 0, 0, 1, 0, 0, sklearn.metrics impatrix = confusion r</pre> | oort confusi | on_matrix | | | | |
| cnf_m | [[98, 9], [18, 29]], dtype= | _ int64) | | | | | |
| from print print print | wates model using makes sklearn import metric ("Accuracy:", metric ("Precision:", metrics ("Recall:", metrics | rics cs.accuracy_ .cs.precisio recall_scor | score(Y_test n_score(Y_te | y_pred)) st, Y_pred)) | and precision | | |
| Precis Recall #This | <pre>cy: 0.824675324675 ion: 0.76315789473 : 0.61702127659574 function plots and t seaborn as cvs</pre> | 68421 47 | the model : | n the form c | f a confusion mat | rix using m | patplot |
| class fig, tick_n plt.x plt.y | _names=[0,1] # name ax = plt.subplots() marks = np.arange() ticks(tick_marks, of ticks(tick_marks, of) | en(class_na class_names) | | | | | |
| cvs.he ax.xa plt.t plt.t plt.y | ate heatmap eatmap(pd.DataFrame xis.set_label_posit ight_layout() itle('Confusion mat label('Actual label label('Predicted label | crix', y=1.1 | | e, cmap="YlG | nBu" ,fmt='g') | | |
| Text(0 | | cted label') n matrix ed label | | | | | |
| - 0 - | 98 | 9 | | - 90 - 80 - 70 - 60 | | | |
| Actual label | 18 | 29 | | - 50 - 40 - 30 | | | |
| г - | ó | i | | - 20 - 10 | | | |
| #Prob | | s import Gau | | del t | | | |
| #Prob #This from gauss gauss | <pre>lem 2 funtion uses the c sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_ d = gaussian.predic</pre> | _ | | | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print | funtion uses the of sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.prediction for the confusion for conf_matrix) | et(X_test) | | | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1 [18 2] print print | funtion uses the of sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.prediction of the confusion of the | atrix to det matrix(Y_tes | t, y_pred) _score(Y_teson_score(Y_t | t, y_pred)) est, y_pred) |) | | |
| #Prob #This from gauss gauss y_pre #Prin cnf_m print [[93 1 [18 2 print print print Accura Precis Recall class tick_i | funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metrics' ('Precision:', metrics' ('Recall:', metrics' cy: 0.792207792207 ion: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange() | atrix to det matrix (Y_tes cs.accuracy rics.precisi s.recall_sco | t, y_pred) _score(Y_teson_score(Y_test, y | t, y_pred)) est, y_pred) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1 [18 2 print print Accura Precis Recall class tick_r plt.s plt.x plt.y ([<mat< td=""><td>funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metrics' ('Precision:', metrics' ('Recall:', metrics' cy: 0.792207792207 ion: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange(fubplots()) ticks(tick_marks, complete to the confusion of the</td><td>atrix to det matrix(Y_tes acs.accuracy rics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca</td><td>_score(Y_teston_score(Y_teston_score(Y_test, y_mes)) mes))</td><td>t, y_pred)) est, y_pred)</td><td></td><td></td><td></td></mat<> | funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metrics' ('Precision:', metrics' ('Recall:', metrics' cy: 0.792207792207 ion: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange(fubplots()) ticks(tick_marks, complete to the confusion of the | atrix to det matrix(Y_tes acs.accuracy rics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca | _score(Y_teston_score(Y_teston_score(Y_test, y_mes)) mes)) | t, y_pred)) est, y_pred) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1 [18 2 print print Accura Precis Recall class tick_n plt.x plt.x plt.y ([<mat< td=""><td>funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metr: ('Precision:', metr: ('Precision:', metrics to the confusion of the</td><td>atrix to det matrix(Y_tes acs.accuracy rics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca</td><td>_score(Y_teston_score(Y_teston_score(Y_test, y_mes)) mes))</td><td>t, y_pred)) est, y_pred)</td><td></td><td></td><td></td></mat<> | funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metr: ('Precision:', metr: ('Precision:', metrics to the confusion of the | atrix to det matrix(Y_tes acs.accuracy rics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca | _score(Y_teston_score(Y_teston_score(Y_test, y_mes)) mes)) | t, y_pred)) est, y_pred) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1 [18 2 print print Accura Precis Recall class tick_n plt.x plt.x plt.y ([<mat <mat="" td="" text<=""><td>funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metrics' ('Precision:', metrics' ('Recall:', metrics' cy: 0.792207792207 ion: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange(fubplots()) ticks(tick_marks, complete to the confusion of the</td><td>atrix to det matrix(Y_tes acs.accuracy rics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca</td><td>_score(Y_teston_score(Y_teston_score(Y_test, y_mes)) mes))</td><td>t, y_pred)) est, y_pred)</td><td></td><td></td><td></td></mat> | funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metrics' ('Precision:', metrics' ('Recall:', metrics' cy: 0.792207792207 ion: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange(fubplots()) ticks(tick_marks, complete to the confusion of the | atrix to det matrix(Y_tes acs.accuracy rics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca | _score(Y_teston_score(Y_teston_score(Y_test, y_mes)) mes)) | t, y_pred)) est, y_pred) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1] [18 2] print print Accura Precis Recall class tick_n plt.x plt.x plt.x plt.x | funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict | atrix to det matrix(Y_tes acs.accuracy rics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca (), 1, '1')]) | _score(Y_teson_score(Y_test), y_mes)) .66e20>, .66a00>], | est, y_pred)) est, y_pred) y_pred)) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1] [18 2] print print Accura Precis Recall class tick_n plt.x plt.x plt.x plt.x plt.x plt.y | funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict | cs.accuracy cics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') | _score(Y_teson_score(Y_test), y_mes)) .66e20>, .66a00>], | est, y_pred)) est, y_pred) y_pred)) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1 [18 2 print print Accura Precis Recall class tick_n plt.x plt.x plt.y ([<mat [text=""]="" td="" text(3<=""><td>funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predicted to the confusion matrix = confusion = co</td><td>cs.accuracy cics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class')</td><td>score(Y_teston_score(Y_teston_score(Y_test, y_test, y_test, y_test, y_test)) 666e20>,66a00>],70</td><td>est, y_pred)) est, y_pred) y_pred))</td><td></td><td></td><td></td></mat> | funtion uses the obsklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predicted to the confusion matrix = confusion = co | cs.accuracy cics.precisi s.recall_sco 7922 11628 47 Len(class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') | score(Y_teston_score(Y_teston_score(Y_test, y_test, y_test, y_test, y_test)) 666e20>,66a00>],70 | est, y_pred)) est, y_pred) y_pred)) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1] [18 2] print print Accura Precis Recall class tick_n plt.x plt.y ([<mat< td=""><td>funtion uses the observation of the sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict</td><td>cs.accuracy cics.precisi s.recall_sco 7922 11628 47 Len(class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class')</td><td>score(Y_teston_score(Y_teston_score(Y_test, y)) mes)) .66e20>, .66a00>], -90 -80</td><td>est, y_pred)) est, y_pred) y_pred))</td><td></td><td></td><td></td></mat<> | funtion uses the observation of the sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict | cs.accuracy cics.precisi s.recall_sco 7922 11628 47 Len(class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class') | score(Y_teston_score(Y_teston_score(Y_test, y)) mes)) .66e20>, .66a00>], -90 -80 | est, y_pred)) est, y_pred) y_pred)) | | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[93 1] [18 2] print print Accura Precall class tick_s plt.x plt.x plt.y ([<mat [text=""]="" td="" text(3)<=""><td>funtion uses the obselvation and sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y) d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metrics cy: 0.792207792207 ion: 0.67441860465: 0.61702127659574 _names = [0, 1] marks = np.arange(interpretation of the confusion matrix in the confusi</td><td>cs.accuracy cics.precisi s.recall_sco 7922 1628 47 Len(class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class') class') class')</td><td>score(Y_teston_score(Y_teston_score(Y_test, Y_test, Y_test, Y_test, Y_test)) .666e20>, .66a00>], .70 .60 .50 .40</td><td>est, y_pred)) est, y_pred) y_pred))</td><td></td><td></td><td></td></mat> | funtion uses the obselvation and sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y) d = gaussian.predict to the confusion matrix = confusion_r(cnf_matrix) 4] 9]] ('Accuracy:', metrics cy: 0.792207792207 ion: 0.67441860465: 0.61702127659574 _names = [0, 1] marks = np.arange(interpretation of the confusion matrix in the confusi | cs.accuracy cics.precisi s.recall_sco 7922 1628 47 Len(class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class') class') class') | score(Y_teston_score(Y_teston_score(Y_test, Y_test, Y_test, Y_test, Y_test)) .666e20>, .66a00>], .70 .60 .50 .40 | est, y_pred)) est, y_pred) y_pred)) | | | |
| #Prob #This from gauss y_pred #Prin cnf_m print [[18 2] print print Accura Precall class tick_s plt.x plt.x plt.y ([<mat [text=""]="" from="" from<="" td=""><td>funtion uses the oblem 3 old cross validation in ticks (tick marks, otticks (tick marks, otti</td><td>cs.accuracy cics.precisi s.recall_sco 7922 1628 47 Len(class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class') class') class') class' cont make_sc</td><td>t, y_pred) _score(Y_test) _score(Y_test) on_score(Y_test) re(Y_test, y) -90 -80 -70 -60 -50 -40 -30 -20 stic regress train_test_</td><td>et, y_pred)) est, y_pred) pred)) e, cmap='YlG sion k =5 fold split, KFold</td><td>nBu', fmt='g')</td><td></td><td></td></mat> | funtion uses the oblem 3 old cross validation in ticks (tick marks, otticks (tick marks, otti | cs.accuracy cics.precisi s.recall_sco 7922 1628 47 Len(class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class') class') class') class' cont make_sc | t, y_pred) _score(Y_test) _score(Y_test) on_score(Y_test) re(Y_test, y) -90 -80 -70 -60 -50 -40 -30 -20 stic regress train_test_ | et, y_pred)) est, y_pred) pred)) e, cmap='YlG sion k =5 fold split, KFold | nBu', fmt='g') | | |
| #Prob #This from gauss gauss y_pred #Prin cnf_m print [[18 2] print print Accura Precall class tick_s plt.x plt.y ([<mat]="" fro<="" from="" td="" text=""><td>funtion uses the obligation = GaussianNB() ian = GaussianNB() ian = GaussianNB() ian fit(X_train, Y_d = gaussian.predict ts the confusion matrix = confusion record (cnf_matrix) 4] 9]] ('Accuracy:', metrics' ('Precision:', metrics' cy: 0.792207792207 ion: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange(subplots()) ticks(tick_marks, conticks(tick_marks, conticks(t</td><td>cs.accuracy cics.precisi s.recall_sco 7922 1628 47 en (class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class') class') class' precision', random_stat (classifier,</td><td>score(Y_testonscore(Y_testonscore(Y_test, Y_test, Y_test</td><td>sion k =5 folds sion k =5 folds split, KFolds late =True) ng=metrics,</td><td>nBu', fmt='g') d , cross_validate ev=cv5, n_jobs=-1</td><td></td><td></td></mat> | funtion uses the obligation = GaussianNB() ian = GaussianNB() ian = GaussianNB() ian fit(X_train, Y_d = gaussian.predict ts the confusion matrix = confusion record (cnf_matrix) 4] 9]] ('Accuracy:', metrics' ('Precision:', metrics' cy: 0.792207792207 ion: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange(subplots()) ticks(tick_marks, conticks(tick_marks, conticks(t | cs.accuracy cics.precisi s.recall_sco 7922 1628 47 en (class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class') class') class') class') class') class' precision', random_stat (classifier, | score(Y_testonscore(Y_testonscore(Y_test, Y_test, Y_test | sion k =5 folds sion k =5 folds split, KFolds late =True) ng=metrics, | nBu', fmt='g') d , cross_validate ev=cv5, n_jobs=-1 | | |
| #Prob #This from gauss gauss y_pred #Prin chrint [18 2 print print [18 2 print [18 2 print print [18 2 print | funtion uses the obselerant players ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict to the confusion matrix = confusion record (cnf_matrix) 4] 9]] ('Accuracy:', metric ('Precision:', metric ('Precision:', metric ('Precision:', metric ('Recall:', metric ('Precision: 0.67441860465 : 0.61702127659574 _names = [0, 1] marks = np.arange(:ubplots()) ticks(tick_marks, confusion matric ('Confusion matric ('Predicted confusion matric | cc.accuracy ccs.accuracy cics.precisi c.recall_sco 7922 1628 47 Len(class_names) class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca c), 1, '1')]) class') class') class') class') class') crt make_sc crt make_sc crt make_sc crt names classifier, Recall: %.4 1: 0.5527 , | recision: (t, y_pred) score(Y_test) score(Y_test) mes)) -90 -80 -70 -60 -50 -40 -30 -20 stic regress train_test_ orer c'recall' e=1, shuffle X, Y, scori f, Precision: (| et, y_pred)) est, y_pred) pred)) et, cmap='YlG sion k =5 fol split, KFolo ate ate ate ate ate ate ate at | nBu', fmt='g') d , cross_validate ev=cv5, n_jobs=-1 | | ']), np.mear |
| #Prob #This from gauss gauss y_pred #Prin cnfint [18 2 print print Accura Precall class tick_s plt.x plt.y ([<mathematy a="" from="" from<="" print="" second="" td="" to=""><td>funtion uses the obligation of the sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict</td><td>ct (X_test) atrix to det atrix to det atrix to det atrix (Y_test) cs.accuracy cis.precisi s.recall_sco 7922 11628 47 cen (class_names) at 0x23c2ca at 0x2ac at 0x2ac at 0x2ac at 0</td><td>## True of the content of the conten</td><td>et, y_pred)) est, y_pred) y_pred)) sion k =5 folk split, KFolk ate =True) ng=metrics, n: %.4f' %(n) .7047</td><td>nBu', fmt='g') d , cross_validate cv=cv5, n_jobs=-1 p.mean(scores['te</td><td>st_accuracy</td><td></td></mathematy> | funtion uses the obligation of the sklearn.naive_bayes ian = GaussianNB() ian.fit(X_train, Y_d = gaussian.predict | ct (X_test) atrix to det atrix to det atrix to det atrix (Y_test) cs.accuracy cis.precisi s.recall_sco 7922 11628 47 cen (class_names) at 0x23c2ca at 0x2ac at 0x2ac at 0x2ac at 0 | ## True of the content of the conten | et, y_pred)) est, y_pred) y_pred)) sion k =5 folk split, KFolk ate =True) ng=metrics, n: %.4f' %(n) .7047 | nBu', fmt='g') d , cross_validate cv=cv5, n_jobs=-1 p.mean(scores['te | st_accuracy | |
| #Prob #This gauss gauss Y_pred #Prin chrint [18 2 print print [18 2 print print Accura Precall class plt.x plt.x plt.y ([$funtion uses the obsklearn.naive_bayesian = GaussianNB() ian.fit(X_train, Y_d = gaussian.prediction in the confusion matrix = cct(X_test) atrix to det atrix to det atrix (Y_test) cs.accuracy cics.precisi crecall_sco 7922 11628 47 cen(class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca co, 1, '1')]) class') class') class') class') class') crimport cort make_sc crimport cort mat, y_pred) score(Y_test) _score(Y_test) _score(Y_test) mes)) .66e20>, .66a00>], .66a00>], .60 -70 -60 -50 -40 -30 -20 stic regress train_test orer cross_valid reclash f, Precision: f,e, cmap='Ylo sion k = 5 fol split, KFolo ate ate True) ng=metrics, a: *.4f' %(n) 7.7047 an: *.4f' %(n) 7.7260 a of Gaussiand , cross_validate cv=cv5, n_jobs=-1 p.mean(scores['te cv=cv10, n_jobs=- p.mean(scores['test_accuracy 1) st_accuracy$ | funtion uses the obsklearn.naive_bayesian = GaussianNB() ian.fit(X_train, Y_d = gaussian.prediction in the confusion matrix = c | ct(X_test) atrix to det atrix to det atrix (Y_test) cs.accuracy cics.precisi crecall_sco 7922 11628 47 cen(class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca co, 1, '1')]) class') class') class') class') class') crimport cort make_sc crimport cort ma | t, y_pred) score(Y_test) _score(Y_test) _score(Y_test) mes)) .66e20>, .66a00>], .66a00>], .60 -70 -60 -50 -40 -30 -20 stic regress train_test orer cross_valid reclash f, Precision: f, | e, cmap='Ylo sion k = 5 fol split, KFolo ate ate True) ng=metrics, a: *.4f' %(n) 7.7047 an: *.4f' %(n) 7.7260 a of Gaussian | d , cross_validate cv=cv5, n_jobs=-1 p.mean(scores['te cv=cv10, n_jobs=- p.mean(scores['te | st_accuracy 1) st_accuracy | |
| #Prob #This from gauss y_pred #Prin causs y_pred #Prin causs y_print [18 2 print print Accurat Print Class plt.x p | funtion uses the obsklearn.naive_bayesian = GaussianNB() ian.fit(X_train, Y_d = gaussian.prediction in the confusion matrix = confusion matrix = confusion in (cnf_matrix) 4] 9]] ('Accuracy:', metricsion:', metr | cs.accuracy cs.accuracy cs.accuracy cs.accuracy cs.accuracy cs.accuracy cs.recall_sco 7922 11628 47 cen(class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca o, 1, '1')]) class_names) class_names) at 0x23c2ca at 0x23c2ca at 0x23c2ca at 0x23c2ca cat 0x23c | score(Y_teston_scor | et, y_pred) est, y_pred) est, y_pred) pred) fine, cmap='Ylog fine, cmap='Ylog fine, k=5 fol split, KFolog fine, k=6 fine) fine, cmap='Ylog fine, | nBu', fmt='g') d , cross_validate cv=cv5, n_jobs=-1 p.mean(scores['te cv=cv10, n_jobs=- p.mean(scores['te | st_accuracy 1) st_accuracy 1 k = 5 | ']), np.mear |
| #Prob #This gauss y_pred #Prin continut [18 2 print print [18 2 print print Accurat Proced Print [18 2 print print Accurat Prob from from from from from from from from | funtion uses the obsklearn.naive_bayesian = GaussianNB() ian.fit(X_train, Y_d = gaussian.prediction in the confusion matrix = confusion matrix = confusion matrix = confusion in (cnf_matrix) 4] 9]] ('Accuracy:', metricsion:', metricsion:', metricsion:', metricsion:', metricsion:', metricsion:', off-41860465' cy: 0.792207792207 ion: 0.67441860465' ion: 0.6742127659574 _names = [0, 1] marks = np.arange(Subplots()) ticks(tick_marks, off-1000 ticks(ti | ct(X_test) atrix to det atrix(Y_test) cs.accuracy cs.accuracy cs.sprecisi crecall_sco recall_sco | score (Y_test) _score (Y_test) _score (Y_test) _n_score (Y_test) fre (Y_test) _ses)) _score (Y_test) _ses) _score (Y_test) _score (Y_test) _ses) _score (Y_test) _ses) _score (Y_test) _score (Y_test) _ses) _score (Y_test) _score | e, cmap='Ylo ie, cmap='Ylo ie, cmap='Ylo ie, cmap='Ylo isplit, KFolo late ieTrue) ng=metrics, in: %.4f' %(n in: %.4f' | d , cross_validate cv=cv5, n_jobs=-1 p.mean(scores['te real validation. g and validation. ecv10, n_jobs=-1) | st_accuracy 1) st_accuracy 1 k = 5 st_accuracy | ']), np.mear |