|  |  |
| --- | --- |
| **EX.NO:3 BAYESIAN INFERENCE** | |
| IN [1] | **import** numpy **as** np  **import** pandas **as** pd  **import** matplotlib.pyplot **as** plt  **import** seaborn **as** sns  **from** mpl\_toolkits.mplot3d **import** Axes3D  **%matplotlib** inline  **from** sklearn.preprocessing **import** StandardScaler  **from** sklearn.decomposition **import** PCA  **from** sklearn.model\_selection **import** GridSearchCV, cross\_val\_score  **from** sklearn.metrics **import** accuracy\_score, confusion\_matrix  **from** sklearn.cluster **import** KMeans  **from** sklearn.linear\_model **import** LogisticRegression |
| IN [2] | y **=** pd**.**read\_csv('/home/vinoth/Downloads/ML exercise/ex3/actual.csv')  print(y**.**shape)  y**.**head() |
| IN [3] | y['cancer']**.**value\_counts() |
| IN [4] | y **=** y**.**replace({'ALL':0,'AML':1})  labels **=** ['ALL', 'AML'] |
| IN [5] | df\_train **=** pd**.**read\_csv(“data\_set\_ALL\_AML\_train.csv”)  print(df\_train**.**shape)  df\_test **=** pd**.**read\_csv('data\_set\_ALL\_AML\_independent.csv')  print(df\_test**.**shape) |
| IN [6] | df\_train**.**head() |
| IN [7] | df\_test**.**head() |
| IN [8] | train\_to\_keep **=** [col **for** col **in** df\_train**.**columns **if** "call" **not** **in** col]  test\_to\_keep **=** [col **for** col **in** df\_test**.**columns **if** "call" **not** **in** col]  X\_train\_tr **=** df\_train[train\_to\_keep]  X\_test\_tr **=** df\_test[test\_to\_keep] |
| IN [9] | train\_columns\_titles **=** ['Gene Description', 'Gene Accession Number', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23', '24', '25', '26', '27', '28', '29', '30', '31', '32', '33', '34', '35', '36', '37', '38']  X\_train\_tr **=** X\_train\_tr**.**reindex(columns**=**train\_columns\_titles) |
| IN [10] | test\_columns\_titles **=** ['Gene Description', 'Gene Accession Number','39', '40', '41', '42', '43', '44', '45', '46','47', '48', '49', '50', '51', '52', '53', '54', '55', '56', '57', '58', '59', '60', '61', '62', '63', '64', '65', '66', '67', '68', '69', '70', '71', '72']  X\_test\_tr **=** X\_test\_tr**.**reindex(columns**=**test\_columns\_titles) |
| IN [11] | X\_train **=** X\_train\_tr**.**T  X\_test **=** X\_test\_tr**.**T  print(X\_train**.**shape)  X\_train**.**head() |
| IN [12] | X\_train**.**columns **=** X\_train**.**iloc[1]  X\_train **=** X\_train**.**drop(["Gene Description", "Gene Accession Number"])**.**apply(pd**.**to\_numeric)  X\_test**.**columns **=** X\_test**.**iloc[1]  X\_test **=** X\_test**.**drop(["Gene Description", "Gene Accession Number"])**.**apply(pd**.**to\_numeric)  print(X\_train**.**shape)  print(X\_test**.**shape)  X\_train**.**head() |
| IN [13] | X\_train **=** X\_train**.**reset\_index(drop**=True**)  y\_train **=** y[y**.**patient **<=** 38]**.**reset\_index(drop**=True**)  X\_test **=** X\_test**.**reset\_index(drop**=True**)  y\_test **=** y[y**.**patient **>** 38]**.**reset\_index(drop**=True**) |
| IN [14] | X\_train**.**describe() |
| IN [15] | X\_train\_fl **=** X\_train**.**astype(float, 64)  X\_test\_fl **=** X\_test**.**astype(float, 64)  scaler **=** StandardScaler()  X\_train\_scl **=** scaler**.**fit\_transform(X\_train\_fl)  X\_test\_scl **=** scaler**.**transform(X\_test\_fl) |
| IN [16] | pca **=** PCA()  pca**.**fit\_transform(X\_train) |
| IN [17] | total **=** sum(pca**.**explained\_variance\_)  k **=** 0  current\_variance **=** 0  **while** current\_variance**/**total **<** 0.90:  current\_variance **+=** pca**.**explained\_variance\_[k]  k **=** k **+** 1  print(k, " features explain around 90% of the variance. From 7129 features to ", k, ", not too bad.", sep**=**'')  pca **=** PCA(n\_components**=**k)  X\_train**.**pca **=** pca**.**fit(X\_train)  X\_train\_pca **=** pca**.**transform(X\_train)  X\_test\_pca **=** pca**.**transform(X\_test)  var\_exp **=** pca**.**explained\_variance\_ratio\_**.**cumsum()  var\_exp **=** var\_exp**\***100  plt**.**bar(range(k), var\_exp); |
| IN [18] | pca3 **=** PCA(n\_components**=**3)**.**fit(X\_train)  X\_train\_reduced **=** pca3**.**transform(X\_train)  plt**.**clf()  fig **=** plt**.**figure(1, figsize**=**(10,6 ))  ax **=** Axes3D(fig, elev**=-**150, azim**=**110,)  ax**.**scatter(X\_train\_reduced[:, 0], X\_train\_reduced[:, 1], X\_train\_reduced[:, 2], c **=** y\_train**.**iloc[:,1], cmap **=** plt**.**cm**.**Paired, linewidths**=**10)  ax**.**set\_title("First three PCA directions")  ax**.**set\_xlabel("1st eigenvector")  ax**.**w\_xaxis**.**set\_ticklabels([])  ax**.**set\_ylabel("2nd eigenvector")  ax**.**w\_yaxis**.**set\_ticklabels([])  ax**.**set\_zlabel("3rd eigenvector")  ax**.**w\_zaxis**.**set\_ticklabels([]) |
| IN [19] | fig **=** plt**.**figure(1, figsize **=** (10, 6))  plt**.**scatter(X\_train\_reduced[:, 0], X\_train\_reduced[:, 1], c **=** y\_train**.**iloc[:,1], cmap **=** plt**.**cm**.**Paired, linewidths**=**10)  plt**.**annotate('Note the Brown Cluster', xy **=** (30000,**-**2000))  plt**.**title("2D Transformation of the Above Graph ") |
| IN [20] | print("Simply predicting everything as acute lymphoblastic leukemia (ALL) results in an accuracy of ", round(1 **-** np**.**mean(y\_test**.**iloc[:,1]), 3), ".", sep **=** '') |
| IN [21] | log\_grid **=** {'C': [1e-03, 1e-2, 1e-1, 1, 10],'penalty': ['l1', 'l2']}  log\_estimator **=** LogisticRegression(solver**=**'liblinear')  log\_model **=** GridSearchCV(estimator**=**log\_estimator,  param\_grid**=**log\_grid,  cv**=**3,  scoring**=**'accuracy')  log\_model**.**fit(X\_train, y\_train**.**iloc[:,1])  print("Best Parameters:\n", log\_model**.**best\_params\_)  best\_log **=** log\_model**.**best\_estimator\_  log\_pred **=** best\_log**.**predict(X\_test)  print('Logistic Regression accuracy:', round(accuracy\_score(y\_test**.**iloc[:,1], log\_pred), 3))  cm\_log **=** confusion\_matrix(y\_test**.**iloc[:,1], log\_pred)  ax **=** plt**.**subplot()  sns**.**heatmap(cm\_log, annot**=True**, ax **=** ax, fmt**=**'g', cmap**=**'Greens')  ax**.**set\_xlabel('Predicted labels')  ax**.**set\_ylabel('True labels')  ax**.**set\_title('Logistic Regression Confusion Matrix')  ax**.**xaxis**.**set\_ticklabels(labels)  log\_pred **=** best\_log**.**predict(X\_test)  print('Logistic Regression accuracy:', round(accuracy\_score(y\_test**.**iloc[:,1], log\_pred), 3))  cm\_log **=** confusion\_matrix(y\_test**.**iloc[:,1], log\_pred)  ax **=** plt**.**subplot()  sns**.**heatmap(cm\_log, annot**=True**, ax **=** ax, fmt**=**'g', cmap**=**'Greens')  ax**.**set\_xlabel('Predicted labels')  ax**.**set\_ylabel('True labels')  ax**.**set\_title('Logistic Regression Confusion Matrix')  ax**.**xaxis**.**set\_ticklabels(labels)  ax**.**yaxis**.**set\_ticklabels(labels, rotation**=**360); |

OUTPUT

|  |  |
| --- | --- |
| OUT [17] | 22 features explain around 90% of the variance. From 7129 features to 22, not too bad. |
| OUT [18] | [Text(-60000.0, 0, ''),  Text(-40000.0, 0, ''),  Text(-20000.0, 0, ''),  Text(0.0, 0, ''),  Text(20000.0, 0, ''),  Text(40000.0, 0, ''),  Text(60000.0, 0, ''),  Text(80000.0, 0, '')] |
| OUT [18] | Text(0.5, 1.0, '2D Transformation of the Above Graph ') |