Divya Thomas

Class: CS 677

Date: 4/24/2023

Non-code answers to assignment.

**Question 1.** Pandas dataframe is printed in question1.py file.

**Question 1-1.** Prediction dataframe printed in question1.py file.

Accuracy: 88.57%

Confusion Matrix:

[[29 2]

[ 6 33]]

**Question 1-2.** Prediction dataframe printed in question1.py file.

Accuracy: 92.86%

Confusion Matrix:

[[30 1]

[ 4 35]]

**Question 1-3.** Prediction dataframe printed in question1.py file.

Accuracy: 85.71%

Confusion Matrix:

[[30 1]

[ 9 30]]

**Question 2-1.** Prediction dataframe for Naïve Bayesian classifier printed in question2.py file.

Accuracy: 91.43%

Confusion Matrix:

[[29 2]

[ 4 35]]

**Question 2-2.** Please see table below for summary of findings

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **TP** | **FP** | **TN** | **FN** | **Accuracy (%)** | **TPR** | **TNR** |
| **Linear Kernel SVM** | 29 | 6 | 33 | 2 | 88.57 | 0.94 | 0.85 |
| **Gaussian Kernel SVM** | 30 | 4 | 35 | 1 | 92.86 | 0.97 | 0.9 |
| **Polynomial Kernel SVM** | 30 | 9 | 30 | 1 | 85.71 | 0.97 | 0.77 |
| **Naïve Bayesian** | 29 | 4 | 35 | 2 | 91.43 | 0.94 | 0.9 |

As you can tell through the summary above, all four models are fairly accurate, especially in calculating true positives (all TPR > 0.9). The Gaussian kernel SVM and Naïve Bayesian, however, were much higher in accuracy for calculating negatives than the others, making them more accurate overall. Between Gaussian and NB, it seems Gaussian wins by about 1% in overall accuracy due to its slightly higher TPR.

**Question 3-1.** See the plot below for the data on k-means clustering on full data with k = 1-8.

Chart, line chart

Description automatically generated

Using the “knee” method, I would say the most optimal k value in this dataset would be 3. This seems to be the spot that the distortion begins to level out without as much of a hard drop. Anything past that value would provide general the same distortion.

**Question 3-2.** Please see the plot below for randomly selected features f4 and f5 after using kmeans clustering on k=3.

Chart, scatter chart

Description automatically generated

As you can see from the plot, it seems that the third cluster (2) has values higher up on the graph, where f4 >6.00 and f5> 3.5. On the other hand, the second cluster (1) has values lower on the graph, where f4 < 5.50 and f5 <3.2. The first cluster (0) is located directly in between the other two. You can see the centroids for each cluster has a linear increase from cluster 1 to cluster 0 to cluster 2.

**Question 3-3.**

Cluster: 0 - Centroid: (3.33, 5.63) - Class: 1

Cluster: 1 - Centroid: (2.89, 5.23) - Class: 3

Cluster: 2 - Centroid: (3.73, 6.24) - Class: 2

**Question 3-4.** See the prediction dataframe on question3.py output.

The accuracy of this classifier on this entire dataset is 83.81%

**Question 3-5.** See the prediction dataframe on question3.py output

Accuracy: 87.68%

Confusion Matrix:

[[52 16]

[ 1 69]]

Analysis: This new classifier, when compared to the others in question 2, is one of the least accurate of the classifiers. It is close in accuracy to the linear kernel SVM (1% less) which makes it more accurate than the Polynomial kernel SVM only. It also has higher number of false negatives than any other classifier tested. Overall, it seems the better options for predicting values for this dataset, based off of the classifiers tested, is still Gaussian kernel SVM or Naïve Bayesian.