Aprendizagem 2024/25

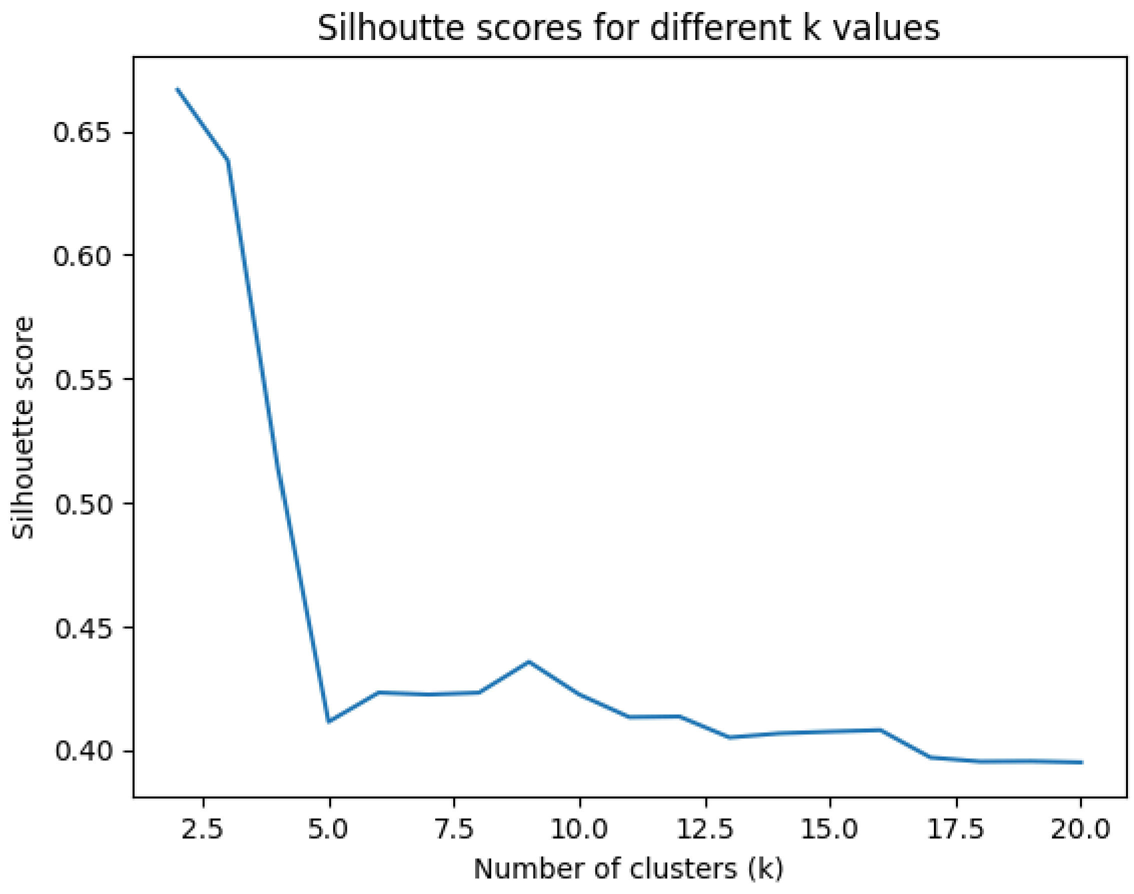
Merit Prize – Group 014

(ist106266, ist106583)

1) The logistic regression model was applied to the training data, achieving an accuracy of 97.7%.

2) The EM clustering algorithm was applied with different numbers of clusters, ranging from two to twenty. Furthermore, the quality of the solution was evaluated using the silhouette score, which measures how well separated and cohesive the clusters are, with values closer to one representing outstanding solutions.

The highest silhouette score of 0.667 was observed with two clusters. As the number of clusters increases, the Silhouette Score drops significantly, stabilizing at about 0.41 from five clusters onwards, indicating that increasing the number of clusters beyond two does not improve the clustering solution's quality. Thus, the optimal number of clusters is expected to be two.

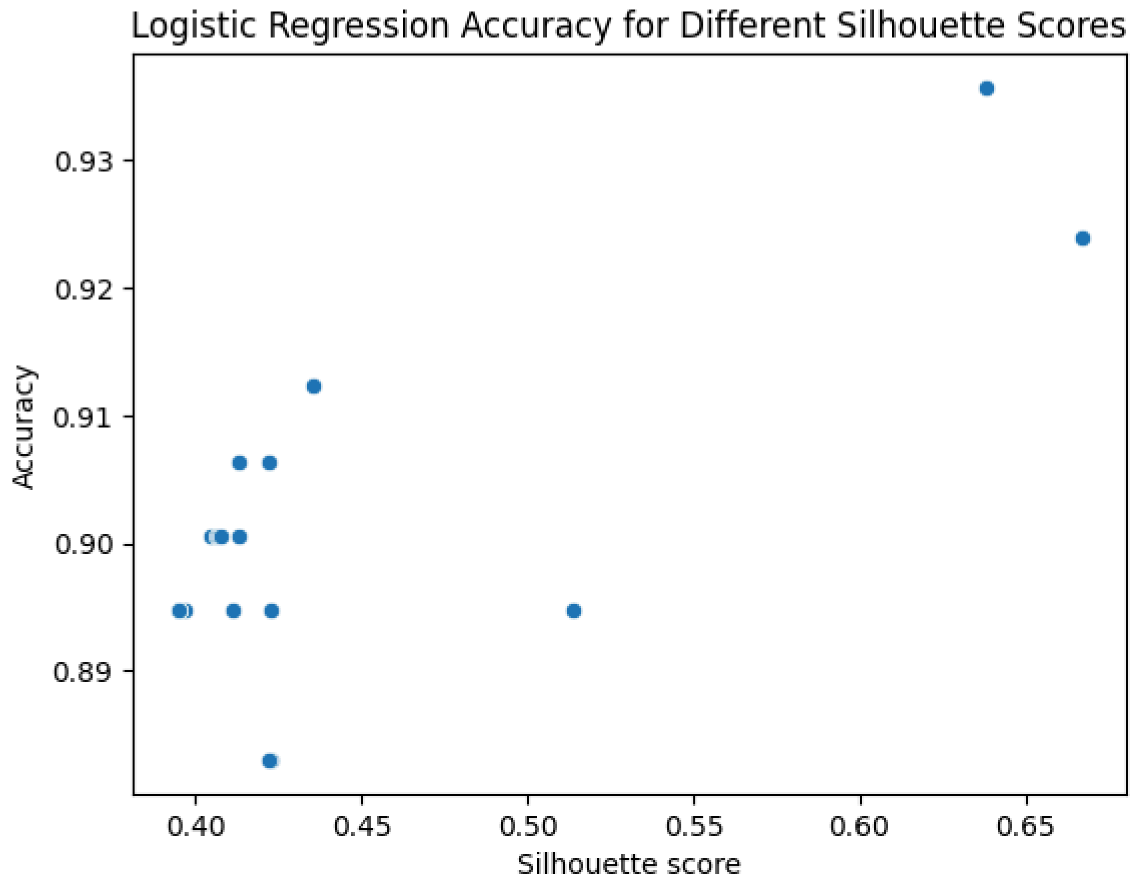


3) The test set was mapped into probability values of the two clusters identified by the EM clustering algorithm. Each data point, originally represented in a high-dimensional feature space, was transformed into a 2-dimensional feature space with each value representing the probabilities of the data point belonging to each cluster.

4) A logistic regression was performed on the test set mapped into a 2-dimensional space of cluster probabilities. The observed accuracy was 92.4%, slightly lower than the one achieved with the original feature set (97.7%).

This drop in accuracy is likely due to the information lost in the dimensional reduction inherent to the mapping process. While in the original feature space, the model could capture subtle patterns across multiple features, in the reduced feature space it relies only on the probability of each data point belonging to the two clusters.

Additionally, by performing logistic regression on the other clustering solutions, a Pearson correlation coefficient of 0.753 was observed, strongly indicating that the better the silhouette, the higher the accuracy of the logistic regression. Surprisingly, however, the three cluster solution seems to outperform the two cluster one in this case.



5) The RBF network, with the hidden layer's RBF parameters learned from the optimal clustering solution with two clusters, achieved an accuracy of only 63.2%, much lower than the 92.4% accuracy obtained from the logistic regression on the mapped data or the 97.7% obtained from the logistic regression on the original feature set. Furthermore, it is important to note that logistic regression was used in the RBF network's output neuron due to the classification nature of the problem at hand.

Aprendizagem 2024/25

Merit Prize – Group 014

(ist106266, ist106583)

6) In general, all results up until the last question were expected. While the logistic regression on the mapped dataset also showed quite an impressive result with an accuracy of 92.4% with only two clusters, this is mostly due to a good fit of the EM clustering process. The RBF network's accuracy, on the other hand, was quite puzzling since, even though the training process was quite similar to question 4, the accuracy was only 63.2%.

Nevertheless, after careful consideration, this seems to be the case mostly due to *sklearn's predict_proba* function normalising the cluster probabilities for each data point. That is, it seems the logistic regression performed on question 4 is very reliant on this normalization step, which is not available to the RBF network.

Let us compare both cases. In the logistic regression on the mapped dataset (question 4), let $n_1(x)$ and $n_2(x)$ be the normalized probabilities of an input x belonging to clusters 1 and 2, respectively. It is possible to write the output, o_1, of the regression as follows:

$$o_1 = \sigma(b + w_1 n_1(x) + w_2 n_2(x)) = \sigma(b + w_1 \frac{c_1(x)}{c_1(x)+c_2(x)} + w_2 \frac{c_2(x)}{c_1(x)+c_2(x)})$$

$$c_i = \frac{\pi_i}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma_i|}}\Phi(x, \mu_i, \Sigma_i),\ where\ \Phi(x, \mu_i, \Sigma_i) = exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i))$$

Note that d is the number of features of our dataset and Φ is, essentially, just a d-dimensional RBF. Now regarding the RBF network's output, o_2, it can be formulated as such:

$$o_1 = \sigma(b' + w'_1\Phi(x, \mu_1, \Sigma_1) + w'_2\Phi(x, \mu_2, \Sigma_2)) = \sigma(b' + w'_1\alpha_1 c_1(x) + w'_2\alpha_2 c_2(x))$$

$$\text{where } \alpha_i = \frac{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma_i|}}{\pi_i}$$

Comparing the two yields:

$$o_1 = \sigma(b + w_1 \frac{c_1(x)}{c_1(x)+c_2(x)} + w_2 \frac{c_2(x)}{c_1(x)+c_2(x)}) = \sigma(b' + w'_1\alpha_1 c_1(x) + w'_2\alpha_2 c_2(x)) = o_2$$

$$\Rightarrow b = b',\ w'_1 = \frac{w_1}{\alpha_1(c_1(x)+c_2(x))},\ w'_2 = \frac{w_2}{\alpha_2(c_1(x)+c_2(x))}$$

Although α_i and w_i are constants (at least after fitting the first model that is), $c_1(x)+c_2(x)$ is clearly not. This way, the first model is essentially the RBF network with an additional non-linear transformation on top. Therefore, it is not trivial for a single output neuron to approximate this change, meaning the benefits this transformation appears to have on accuracy are not non-existent in the RBF network, decreasing the quality of its classification capabilities. Moreover, by merely normalizing the RBF outputs in the RBF network without even taking into account other factors such as α_i, the model's accuracy rises from 63.2% to 91.8%, almost matching question 4's logistic regression model.

Note: in this homework, random_state=42 was used whenever possible to ensure a fair comparison between different hyperparameterizations.

END