Lab1 Report on the k-NN classifier

Experiment 1:

In this experiment, the k-NN classifier was evaluated using both Euclidean distance and 1 - cosine similarity by randomly selecting 80 articles per class for training and using the remaining articles for testing. Calculation of the accuracy was done of the k-NN classifier on the testing samples by setting k to 5. This procedure was repeated 20 times to obtain a more reliable accuracy estimate.

Finding:

- The Euclidean distance-based k-NN classifier obtained 0.87 +- 0.03, and the 1-cosine distance-based k-NN classifier received $0.96 +- 0.01. The reason for a low accuracy rate of the Euclidean-based k-NN is due to several reasons:
- Euclidean distance measures the straight-line distance between 2 data points in Euclidean space. The magnitude or size of the data points heavily influences this distance metric. For example, when comparing two vectors with different sizes, one with a large number and one with a small number, the Euclidean distance would be considerable because they have different sizes.
- However, the cosine distance measures the angle between these 2 data points. It focuses on the orientation instead of the magnitude. If the two vectors point in the same direction, they are the same. If they are pointing in different directions, they are not the same. It is better when we want to consider the direction of the two vectors that have different magnitudes. Hence cosine distance performs better than Euclidean distance, especially when dealing with sparse data, like text data in our experiment.

Experiment 2

Experiment 1 proved that cosine similarity performs better, especially with sparse data. In this, the performance of the k-NN cosine similarity-based classifier will be examined for various values of k ranging from 1 to 50.

Method:

80 articles per class were randomly selected for training and the remaining articles were used for testing. The range of k values was from 1 to 50, and the procedure was repeated 20 times.

Discuss in your report the difference between the training and testing accuracies and why this is the case. Analyse in your report the effect of k based on this experiment. What do you think is a reasonable value for k?

- This experience indicates that there is a slight difference between training accuracy and testing accuracy. The training accuracy here is greater than the testing accuracy for all values of k (1-50). This behaviour is called overfitting. This happens when the model learns the training data too well and fails to generalise to new, unobserved data. This suggests that the model is not doing well on new data, a crucial indicator for judging the model's effectiveness. To make the model perform well on new data and prevent overfitting, we need to choose the most suitable value for k.
- **Analysing the behaviour of k:** The model has a low bias but significant variance at small values of k. This is caused by overfitting. For large values of k., the bias increases and the variance decrease. This is because assigning large k neighbours to the model creates a low variance.
- **Choosing the correct value for k:** Choosing a small value for k, like 1 would cause overfitting and choosing a large value of k would cause an increase in bias. A good value for k would be in the range of 7-10. This balances the variance and bias. It can

be concluded from the experiment that the k-NN algorithm can be significant;y impacted by the k value.

Findings:

1. The number of neighbours employed to classify data considerably affects the k-NN classifier's performance. By using this distance metric, it was concluded that the best value of k is around 7, which had the lowest average testing error rate and the slightest standard deviation:

- training error rates: 0.02 ± 0.01
- testing error rates: 0.04 ± 0.01

2. Low training and high testing errors are common for low values of k because the model overfits the training data. But also, the model can overfit the data for the high value of k, which can lead to high training error rates and high testing error rates.This difference between the training and testing accuracies is known as the bias-variance trade-off. In conclusion, a 1-cosine similarity is an ideal approach in text classification tasks when choosing the correct value of k.

Experiment 3

What classes do you think these 5 articles should belong to, based on your judgement of their content? Can your classifier make an appropriate class prediction for these 5 articles? Analyse the reason for your answers in your report.

- I don't believe that any of these 5 articles belong to any of the 4 classes.
- The k-NN classifier is trained based on the data that is provided on our training data set, which includes the 4 classes and the 200 articles that are associated with that class. Based on the result of Experiment 2, when k has a low value, a value of 3 in this case, it will cause the problem of overfitting and an increase in the variance. This is caused by the model classifying the text based on the limited number of neighbours. This results in the classifier not being able to make an appropriate class prediction for these 5 articles.

Comment on your classifier's performance in your report. What are the consequences of having no training data and limited training data for the 'sports' class?

- It is expected that the "sports" classifier will perform worse than the other classes in this category because there are only three training examples for this class. Due to the classifier's limited exposure to the sports class, it is more difficult to correctly categorise fresh samples because it has fewer data to work with. In general, bias models of some classes, such as the sports class in our experiment, can originate from data that is unbalanced and includes very few samples.

Self-learn the concepts of zero-shot learning and few-shot learning. In your report, link these concepts to the experiments you've just performed. Is your model performing zero- or few-shot learning? Explain your reasoning.

- Zero-shot learning enables the model to recognise new objects that it has never seen before. This happens by giving the model some information about these objects. During classification, when the model encounters something it has never seen before, it utilises the information that is already in its data to classify the object.

- However, few-shot learning is a type of learning where the model in trained to recognise new objects or classes using only the limited training data provided. This model in this learning is able to learn new concepts from the small amount of data provided by employing transfer learning techniques.

- In this experiment, the model uses few-shot learning. This is due to the fact that only 5 articles are being used to train the model to be able to classify the "sport" class. The model uses transfer learning techniques to be able to this new "sport" class from the limited samples given.

Choose a training-testing trial in Experiment 2 for k=1. Observe the testing error of this 1-NN and estimate the interval where its true error lies with 90 % probability. Explain in your report how you compute it.

- The confidence interval in a range of numbers that helps us determine the performance of our classifying algorithm. It gives us a representation of how much the computer's performance is when we run the experiment multiple times, each time choosing a random set of text to classify. In our experiment, we are testing the range of the computer's error rates based on the outcome of the classification algorithm. The upper bound and lower bound are computed to determine the best-case scenario and the worst-case scenario that the test error lies between. So the 90 confidence interval lies between the upper bound and the lower bound. This means that when repeating the procedure many times, the test error of the experiment falls between the upper bound and lower bound.

Method:
For Experiment 2, a training-testing trial was selected, and k was set to 1. A total of 200 samples were sampled for each of the 4 classes, with 80 samples from each class being used for training and the remaining samples for testing. The true testing error for the 1-NN classifier was then calculated for a 90% probability. A z-score of 1.64 was chosen for the 90% probability. Finally, the confidence interval was computed and the lower and upper bounds as well. This helped determine the true testing error of the 1-NN classifier which falls within the range of the upper and lower bound.