COMPSCI 361 - Assignment 3 Report

Ian Chang, Adam Ghafouri, Matheson (Matt) James, Kevin Jiang, Xingjie (Jason) Zheng

Task 1

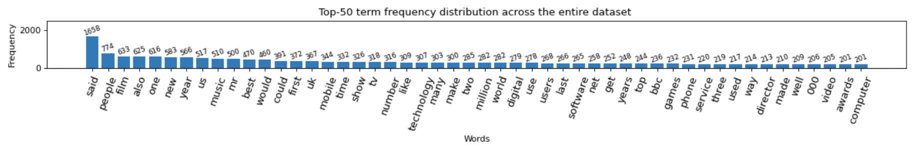
Part a

The dataset contains 534 articles in total. From this dataset, 14927 features/words were extracted. The following shows 5 example articles with the extracted features.

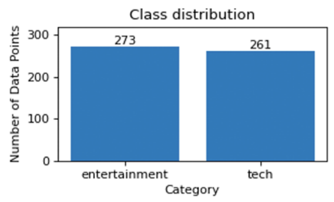
```
   ArticleId       Category  00  000  000th  001st  0051  007  0100  0130  ...  zombie  zombies  zone  zonealarm  zones  zoom  zooms  zooropa  zorro  zutons
0       1976           tech   0    1      0      0     0    0     0     0  ...       0        0     0          0      0     0      0        0      0       0
1       1797  entertainment   0    0      0      0     0    0     0     0  ...       0        0     0          0      0     0      0        0      0       0
2       1866  entertainment   0    0      0      0     0    0     0     0  ...       0        0     0          0      0     0      0        0      0       0
3       1153  entertainment   0    0      0      0     0    0     0     0  ...       0        0     0          0      0     0      0        0      0       0
4        342  entertainment   0    0      0      0     0    0     0     0  ...       0        0     0          0      0     0      0        1      0       0

[5 rows x 14929 columns]
```

Part b

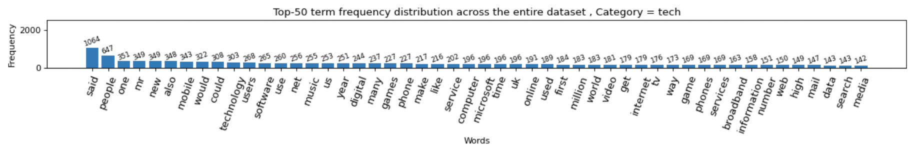
(i) - Plot for the top 50 term frequency distribution across the entire dataset

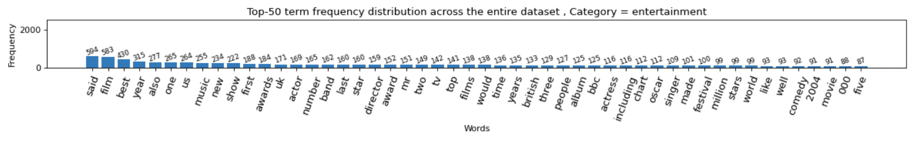


(iii) - Plot of the class distribution



(ii) - Plot of the term frequency distribution for respective class of articles





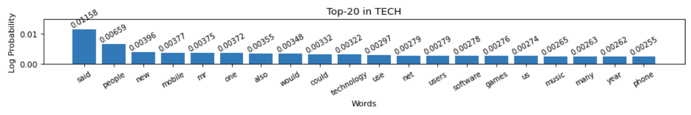
Task 2

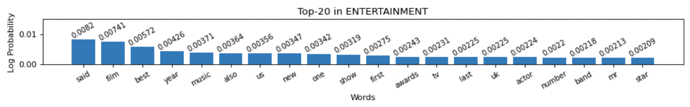
Part a

(i) - The top 20 most identifiable words that are most likely to occur in the articles are shown below.

For the **TECH** class, they are (from most to least):
said › people › new › mobile › mr › one › also › would › could › technology › use › net › users › software › games › us › music › many › year › phone

For the **ENTERTAINMENT** class, they are (from most to least):
said › film › best › year › music › also › us › new › one › show › first › awards › tv › last › uk › actor › number › band › mr › star

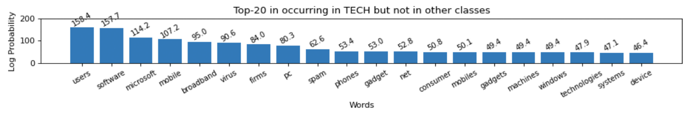


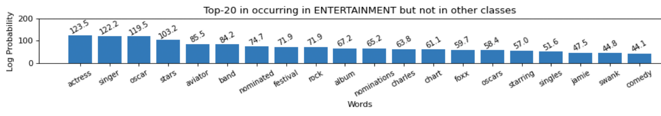


(ii) - The top 20 most identifiable words that maximize $\frac{P(X_w = 1 \mid Y = y)}{P(X_w = 1 \mid Y \neq y)}$ are shown below.

For the **TECH** class, they are (from most to least):
users › software › microsoft › mobile › broadband › virus › firms › pc › spam › phones › gadget › net › consumer › mobiles › gadgets › machines › windows › technologies › systems › device

For the **ENTERTAINMENT** class, they are (from most to least):
actress › singer › oscar › stars › aviator › band › nominated › festival › rock › album › nominations › charles › chart › foxx › oscars › starring › singles › jamie › swank › comedy





In the list generated by part (i), we can see that words like ‘said’, ‘also’ and ‘mr’ are considered. However, these words are not useful for classifying the type of article as they are commonly used words in general regardless of the context.

On the other hand, for the list generated by part (ii), we can see the list of words is more unique to each category. For instance, if we see words like ‘actress’, ‘singer’ and ‘oscar’, we can expect the article to be about entertainment. Likewise, words like ‘users’, ‘software’ and ‘microsoft’ is most relevant to technology.

Thus, the list from part (ii) describes the two classes better.

Part b

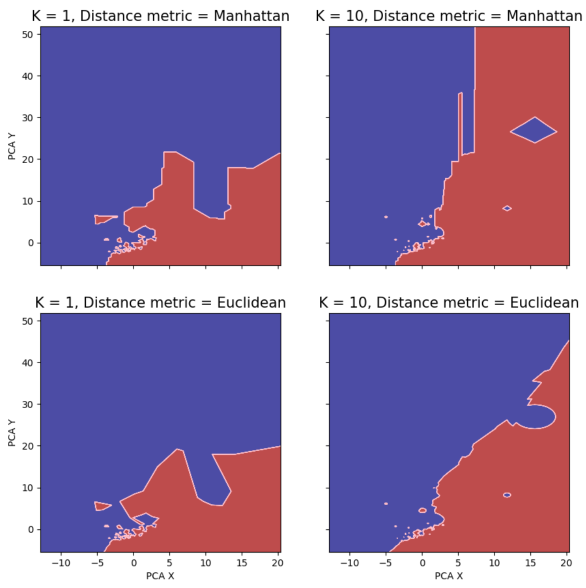
A surface plot for a K nearest neighbours classification model shows the regions in which unseen samples will be classified. Each region can pertain to a single class or multiple regions may be of the same class. So the hyperparameters will affect the shape of the regions as models using different hyper parameters will predict the class of samples differently.To create a surface plot the data must be dimensionally reduced such that it can be shown on a graph. For the purposes of this exercise, it will be dimensionally reduced down to 2d as that is easier to interpret than 3d. PCA was used to reduce the data from over 1000 columns to 2. This could then be plotted as a scatter graph as seen to the right where red points are tech and blue points are entertainment.

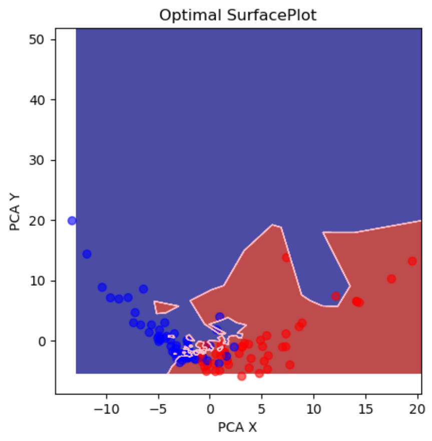
From this, we can create a surface plot denoting the prediction of the model given x and y values by colour. Furthermore, the model highlights the impact of changing the hyper parameters (number of neighbours and distance metric). As seen to the left, The blue regions are tech and entertainment regions are entertainment with a pink decision boundary.

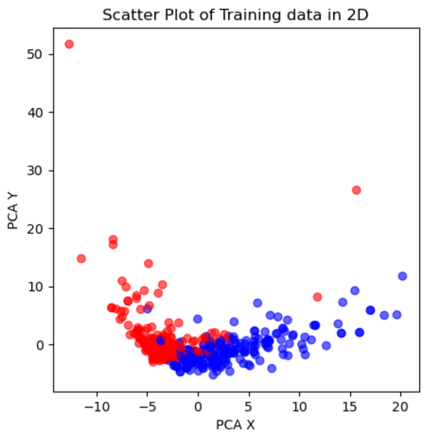
From these, we can see the impact of changing the number of neighbours and how distance is measured. Looking at the impact of the number of neighbours first. By increasing the number of neighbours the decision boundary smoothes out as individual points have less individual influence on the classification of other points. This is highlighted around the 0,0 point where because there is a cluster of points as seen in the scatter plot the decision boundary is more erratic. However, we see this region smoothed slightly when k is increased from 1 to 10. Additionally, we see the influence of outliers such as the point at 15,27 is reduced as the number of neighbours increases, as indicated by the reduction in the surrounding region in the Manhattan surface plots. However, this also means that points may be influenced by irrelevant points at higher k values. The impact of using different distance metrics is especially visible. Under the Manhattan method, there are steeper edges and drop-offs towards the extremities of the model. Additionally, the Manhattan method produces squarer shapes generally due to measuring the combined horizontal and vertical distance and not the length of the direct path. On the other hand, the Euclidean method produces more rounded shapes as indicated by the nodule around the point at 15,27 being rounder in Euclidean and a diamond with Manhattan. Lastly, with Manhattan, we don't see the drop-offs at the extremities and generally produce a smoother curve.

The surface plot given the best hyper params as discussed later is on the right, with a scatter plot of the test data over the top where blue points are tech and red points are entertainment. So where the colors are the same there would be a correct prediction.

Comparison of Surface Plots given different Hyper Parameters

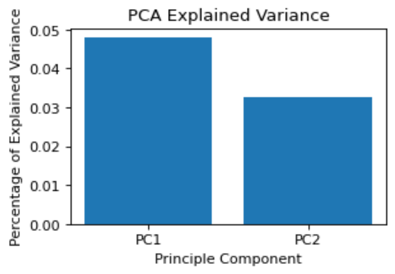






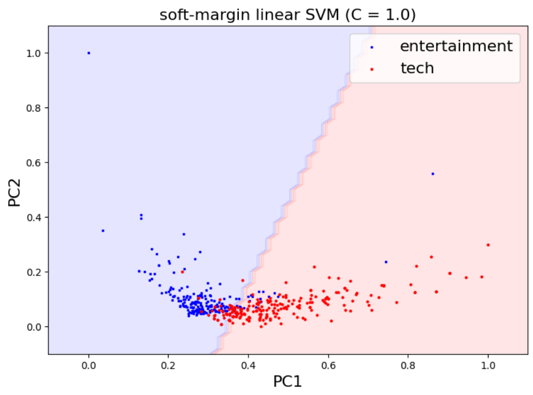
Part c

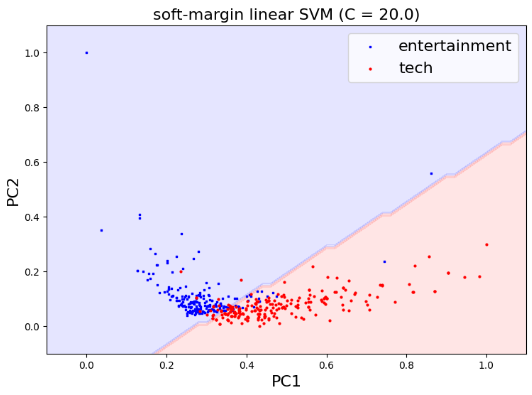
First, we applied principle component analysis (PCA) as the dimensionality reduction algorithm. The following shows the variance for each component.



When selecting a small value of principal components for the PCA procedure, such as 2, it leads to a limited explanation of variance (around 5%). This indicates that a significant amount of information is lost during the dimension reduction process.

(i) - Surface plots for soft margin linear SVM with penalty C

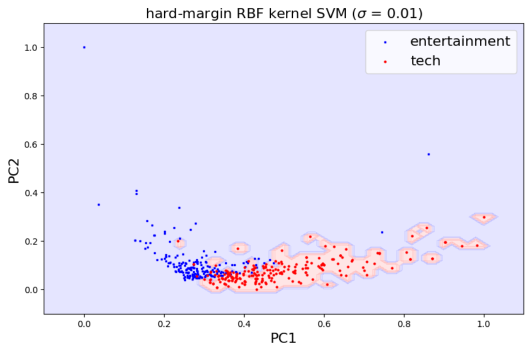


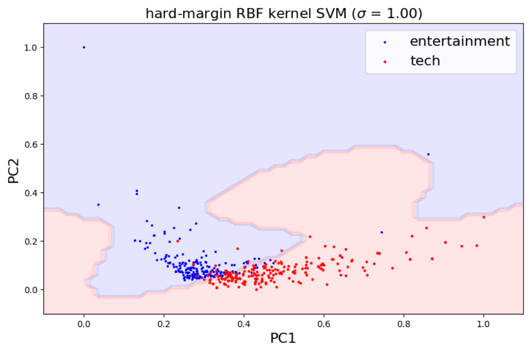


A lower value of C allows for more misclassifications. The classifier becomes less sensitive to individual data, and the margin becomes wider, resulting in a possibly underfitting model.

On the other hand, a higher value of C imposes a stricter penalty on misclassifications, this means that the classifier will try to classify the training data as accurately as possible. As a result, the margin becomes narrower and the decision boundary becomes more sensitive to individual data points, resulting in a possibly overfitted model.

(ii) - Surface plots for hard margin RBF kernelised SVM with kernel width σ





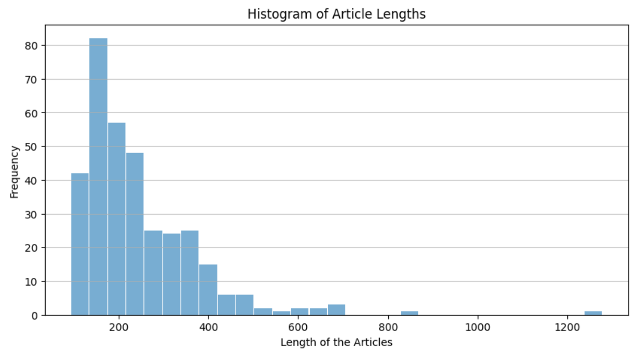
High σ values mean 'wider kernel', the decision boundary becomes smoother, resulting in a possibly underfitting model.

Low σ values mean 'narrow kernel', the decision boundary becomes jagged like the nearest neighbour, resulting in a possibly overfitting model.

Part d

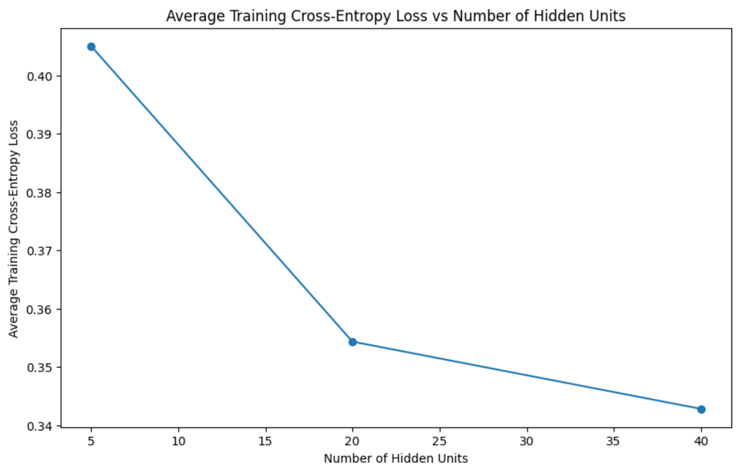
First, we explore the dataset to see what the average lengths of the articles are.

```
+--------------------+--------+
|     Statistic      | Value  |
+--------------------+--------+
|        Min         |   94   |
|    1st Quartile    | 155.25 |
|       Median       | 210.0  |
|    3rd Quartile    | 301.0  |
|        Max         |  1277  |
| Standard Deviation | 130.26 |
+--------------------+--------+
```



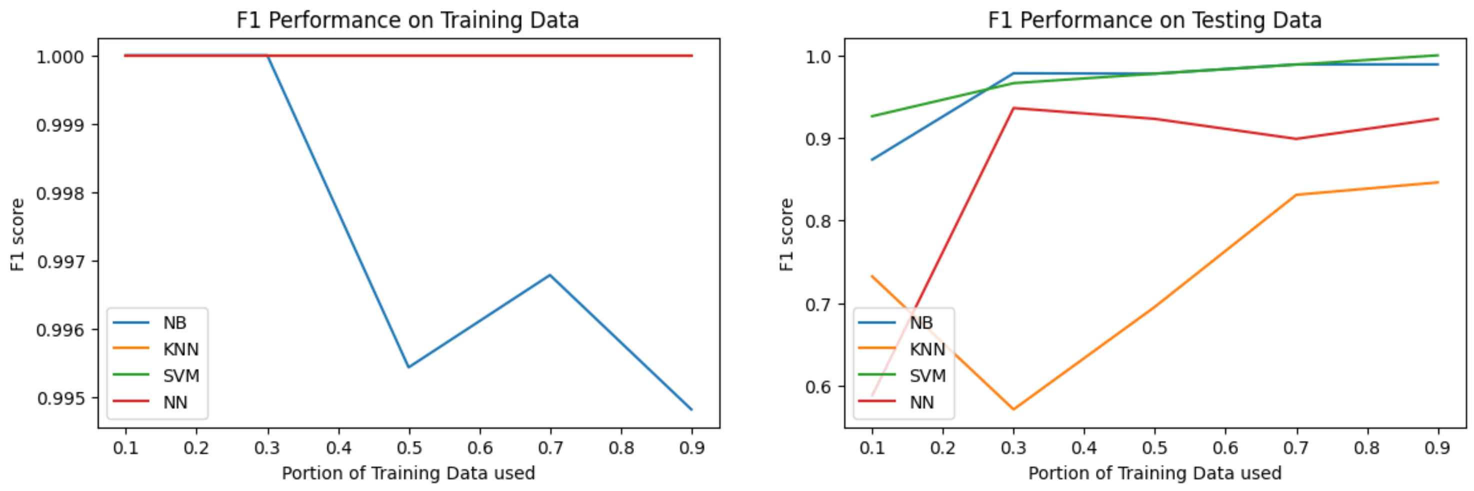
Given these results, the input text has been normalised to 300. So articles longer than 300 words are cut down to 300 words, and articles less than 300 words are padded to have a length of 300 words.

For the training phase, we trained three different neural networks with varying numbers of units of its single hidden layer (i.e. among 5, 20 and 40). The following plot shows how the average training cross-entropy loss changes.



Task 3

Part a

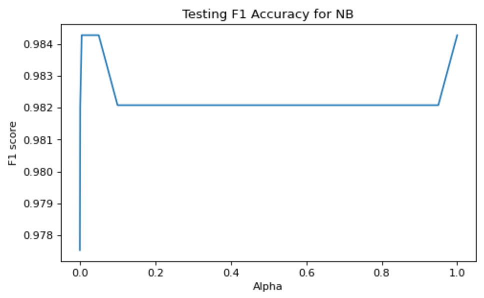


As the portion of training data increases, the F1 score on the testing data seems to increase as well.

Part b

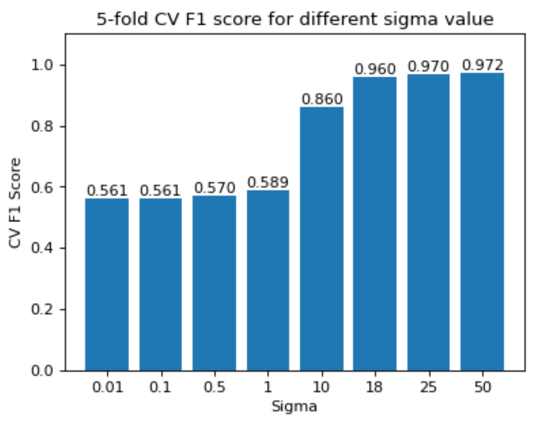
- Naive Bayes

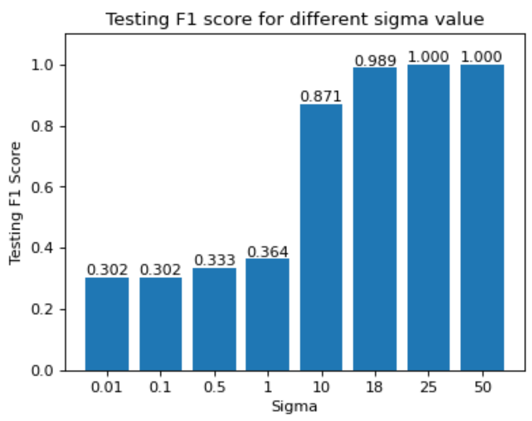
The model was tested by varying the value of alpha, which is the hyperparameter for Laplace smoothing. The values of alpha tested were: 0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.2, 0.5, 0.7, 0.9, 0.95 and 1. Their corresponding F1 scores after 5-fold cross-validation are: [0.97753, 0.98203, 0.98427, 0.98427, 0.98427, 0.98208, 0.98208, 0.98208, 0.98208, 0.98208, 0.98208, 0.98427]



- SVM

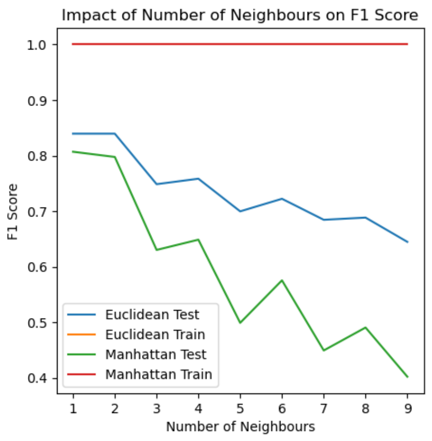
Shown below, as the sigma_value increases from 0.01 to 25, both the CV accuracy and the testing accuracy exhibit an upward trend, the most notable improvement occurring within the range of 1-10, with the highest increase observed. This suggests that the optimal sigma value lies around 25.





- KNN

KNN has two hyper parameters the number of neighbours and the distance measure. The number of neighbours determines the number of points considered in the prediction of an unknown point. As the number of neighbours increases the chance of considering irrelevant points increases which could result in underfitting. On the other hand, as the number decreases the model will likely overfit. The distance measure can be Manhattan or Euclidean, Manhattan measures the combination of horizontal and vertical distance, whilst the Euclidean distance is the direct distance using Pythagoras. The graph on the left shows how the f1 score, with 5-fold cross-validation, varies with the number of neighbours across the two distance measures.



The obvious observation from the graph is the f score for the training results there appears to be only a line however both of them are overlapping with f scores of 1 for all k values. This is just inherent to the way knn works as if you input a training data point it will sit on top of a point in the model hence it will be the closest. As n increases the prediction is weighted based on proximity, so the value from which it is derived will have a weight of 1. Looking at the test results we can see that as k increases the f score decreases. So a low k value will produce the highest f score, both k =1 and k = 2 produce equal k scores Additionally looking at the distance metric euclidean produces a better f score for all values of k. Hence the optimal distance measure is Euclidean.

- NN

Note here that I did not change the optimizer, which is still Adam. I did not change the activation functions for the hidden layers or the output layer. I also did not change the output dimension in the embedding layer, or the cutoff score for deciding whether an article with probability p of being about 'Tech' is labelled as 'Tech' or not. These are further hyper-parameters that could be tweaked and experimented with.

However, earlier in my code, I did choose the number of epochs for training based on the accuracy vs epoch plot. And the length of the sequence vectors based on the histogram and summary statistics.

Also, note that the best hyper-parameter combination will vary somewhat depending on small random perturbations. That is, the values of the best hyper-parameters chosen can vary slightly every time the above code for the 5-fold cross-validation is run.

Part c

- Naive Bayes

Choosing an alpha value of 0.005 seemed to do the best. With this hyperparameter setting, it scored an F1-score of 0.98901

- SVM

Choosing a sigma value of 25 seemed to do the best. With this hyperparameter setting, it scored an F1-score of 1.0

- kNN

The most optimal distance measure for kNN is the Euclidean metric. Furthermore, choosing k to be 1 or 2 seemed to produce equally optimal results. With these hyperparameter settings, it scored an F1-score of 0.8397

- NN

Based on results from the above, the most optimal hyperparameter was:

num_dense_layers: 1 num_units: 60 learning_rate: 0.1 batch_size: 32

In conclusion, the RBF kernelized SVM model is the best model at classifying the news articles into tech and entertainment with an f score of 1.0. This report highlights the importance of model selection when approaching a classification style problem. As well as the impact of tuning hyper parameters and how each hyper parameter impacts each model. Furthermore, it allowed us to explore the differences between how each model performs on test and training data as well as the amount of data each model requires to be effective.