Homework 2

Lawrence Leymarie

Table of contents

1 Analysis

I implemented two classification models for this project to predict whether an email was spam. The dataset that was used was composed of 4601 rows and 58 columns. Of these 58 columns, 57 are continuous predictors for both models, the outcome variable 'Class' binary 0 if the email is not spam and one if it is indeed spam. When looking into the class distribution, it is essential to note that negative cases (0) outnumber positive cases (1) from 60.6% to 39.4%, which means we should expect the models to favor classifying negative cases accurately. Creating a correlation matrix also helps us identify relationships between variables; we can see in the plot below that there are around 13 variables with high correlation, which could cause multicollinearity problems in the future. Still, I will ignore it as it will not significantly impact the models. Finally, upon observation, we can notice the target variable has a higher correlation to words like 'your,' 'remove,' and '$.'

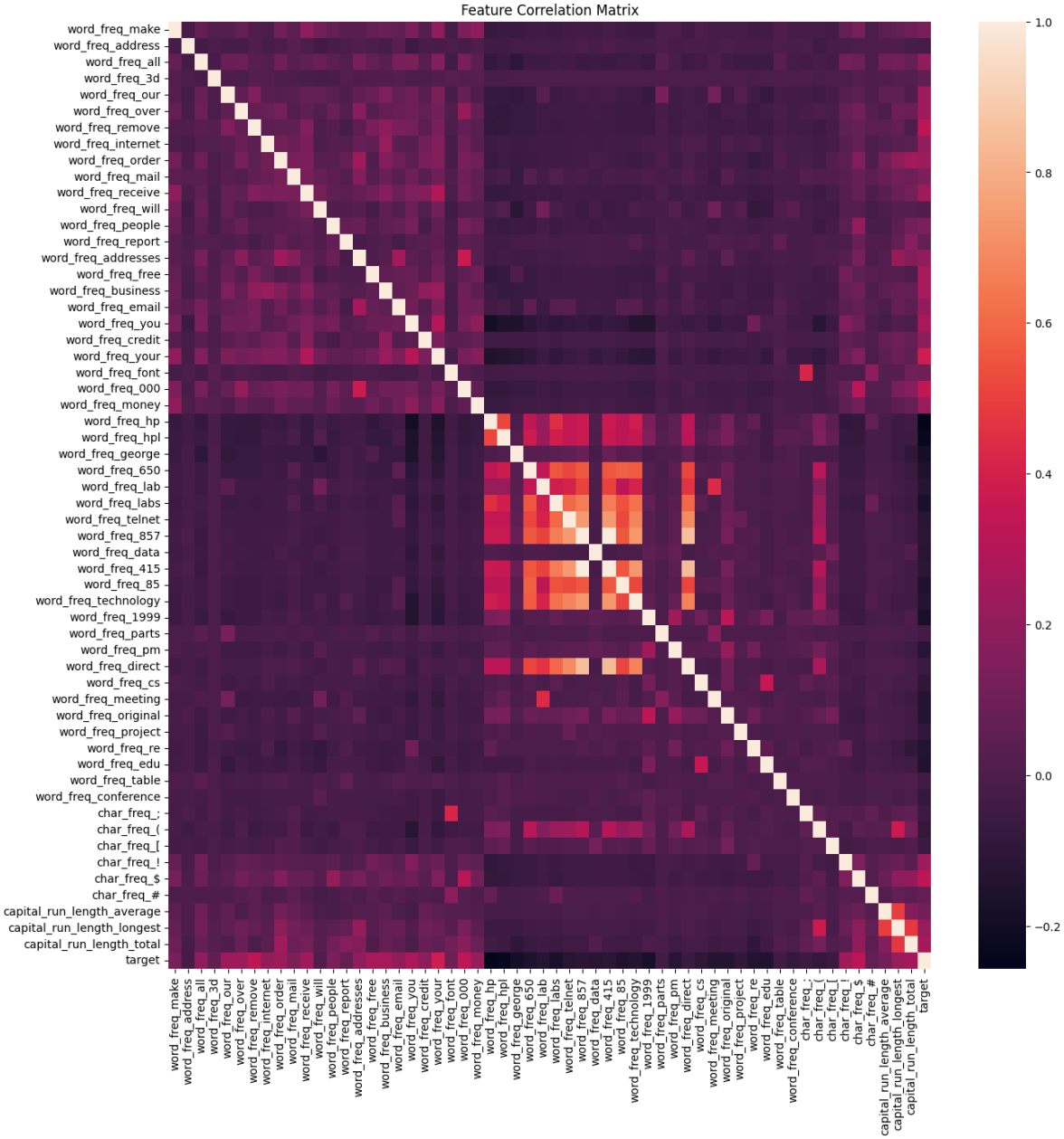

Figure 1: Correlation matrix for all variables

2 Methods

The data was acquired through the 'ucimlrepo' package and split into two data frames: X, containing the predictors, and y, which holds the outcome 'Class.' The predictor column names were stored into a 'preds' variable, which was used to access numerical values to perform z-score transformation after splitting our data into validation (20%) and training (80%)—the next step after preprocessing was to implement the first model, a neural network. The network structure was chosen through extensive testing, starting from a simple overfit model, adding a dropout for every dense layer and an elastic net on every dense layer with a relu activation function. Since this was a classification problem, the last layer comprised one node with a sigmoid activation function. The loss function chosen was binary cross-entropy, which is the most appropriate for a classification problem. Early stopping was also included to avoid overfitting in the late training epochs using loss as the monitor value. Finally, the model was fit with 100 epochs, a batch size of 32, which gave the model its best result.

The logistic regression was implemented using the scikit learn package, which used the training and testing scaled data from the previous section.

3 Results

As was expected, the neural network showed excellent performance on early stopping; we can observe a loss of 0.1402, an accuracy of 96.14% for training, and a loss of 0.1567 and an accuracy of 96.09% for testing. These numbers indicate that the model is good at making predictions with similar metrics between testing and training, showing no signs of overfitting. Using model history, I found the lowest loss achieved was around 0.1520. On the other hand, using the same test train split, the logistic regression had similar results with a loss of 0.2145 and an accuracy of 92.61% during the training phase. During the testing phase, the loss dropped to 0.1824, while the accuracy jumped to 93.70%. Overall, the logistic regression performed slightly worse than the neural network, with a difference in loss of 0.0259 and a decrease in accuracy of 2.39%; these figures are not very significant and indicate that either model could be used for such a task.

4 Reflection

Altogether, although the neural network had a better performance in both loss and accuracy, the logistic regression had a more than acceptable performance. What should be considered for commercial use is that the neural network is the right choice as it will always perform better, and the edge it has compared to the logistic regression is valuable. However, for casual use, it could be argued that using a neural network to predict spam emails is extreme when a logistic regression can achieve similar performance while taking significantly less time to

set up and using less computation power. The benefit of using a neural network is that it works with large datasets and detects complex nonlinear relationships between dependent and independent variables. In the context of this dataset, it is clear that the data is too simple and explains enough correlation so that the logistic regression can perform very closely to the neural network.