Machine Learning Project Report

Imperial College London
Orion Mathews

1. Spam emails

1.1. The Spam problem and solution

In modern society it has become normal to receive an extensive amount of Spam emails. Spam emails are not needed or wanted and receiving such a large volume of them wastes time and bandwidth. Users must be careful not to open or download anything from Spam emails as many contain malware and viruses. Important messages become exceedingly hard to find, impacting the individuals productivity and efficiency.

The solution is to filter out the Spam leaving only the important emails behind. For this to be achieved it is necessary to be able to predict weather an email is Spam or not simply from its contents. This is no simple problem since it is impossible to mathematically model all possible Spam emails. Where sufficient data exists and mathematical models do not, machine learning becomes essential and allows for the construction of such a filter. The aim is to train a machine learning model using known Spam or not Spam email contents so that the program can predict whether future emails are Spam or not Spam.

1.2. The data

The data set has been provided by the UCI Machine Learning Repository: `https://archive.ics.uci.edu/ml/datasets/spambase`. The Spambase Data Set consists of 4601 emails, of which 1813 (39.4%) are Spam. There are 57 features used for detecting Spam, 48 of which count the frequency of various words (as a percentage of total words in each email), a further 6 count the frequency of various characters (as a percentage of total characters in each email) while the last 3 are measures of capital letter run length. The input data is difficult to visualize due to the large number of features. It is impractical to represent all the input data on a single plot. Here 1 is defined as Spam and 0 as non-Spam.

The type of learning is supervised learning as the model is being trained on known input and output data with the aim of predicting future outputs.

2. Baseline classifiers

2.1. Logistic regression

Logistic regression is a machine learning method for classifying data into discrete outcomes. Logistic regression is a linear classifier as the prediction can be written in terms of $\boldsymbol{w^T x}$, which is a linear function of $\boldsymbol{x}$ (where $\boldsymbol{x}$ is the input data and $\boldsymbol{w}$ is a vector of weights). The prediction (or hypothesis) function for logistic regression is defined in equation 1 and gives the probability that the output is equal to 1 (email being Spam).

$$\theta(\boldsymbol{w^T x}) = \frac{1}{1 + e^{-\boldsymbol{w^T x}}} \tag{1}$$

If the hypothesis function gives a number that is greater than 0.5 then the email will be classified as Spam, otherwise the email will be classified as non-Spam.

During the training of the logistic regression model the data was split into 10 disjunct sets for use in cross validation. Each set was spit into training data and test data. This 10-fold cross validation protects against over fitting.

This particular model achieved an accuracy of 92.7%. The confusion matrix in figure 1 summaries the performance of logistic regression in predicting Spam. As shown, Logistic regression predicts Spam correctly 89% of the time, while predicting non-Spam correctly 95% of the time. However it incorrectly predicts non-Spam emails as Spam emails 5% of the time. Figure 2 shows the receiver operating characteristic (ROC) curve for logistic regression.

Logistic regression provides a baseline to which the performance of other models can be compared to. Marking good mail as Spam is particularity undesirable, therefore the aim would be to reduce the false positive rate while maintaining a high true positive rate. In logistic regression, for the false positive rate to be approximately 0% the true positive rate would be around 60%. Meaning that to ensure almost no non-Spam email is predicted as Spam, 40% of Spam emails would not be filtered.

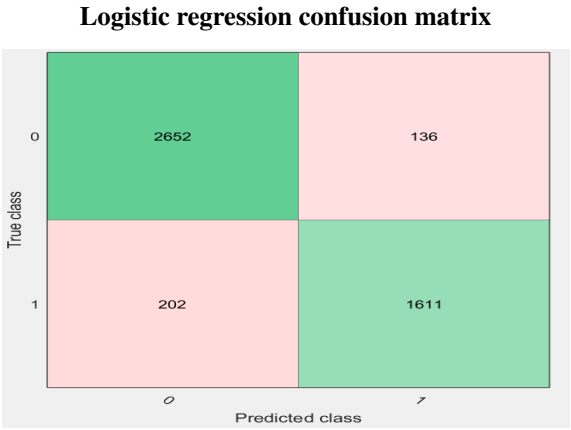

Figure 1. Logistic regression confusion matrix.

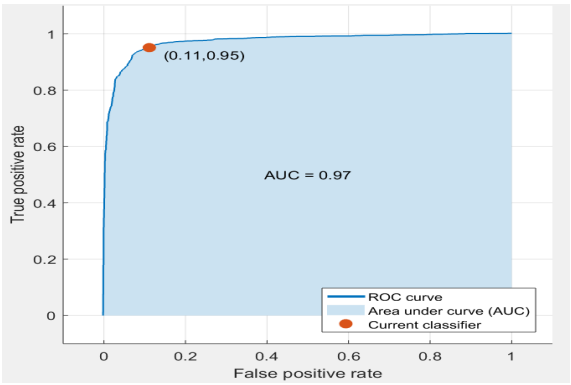

Figure 2. Logistic regression ROC curve

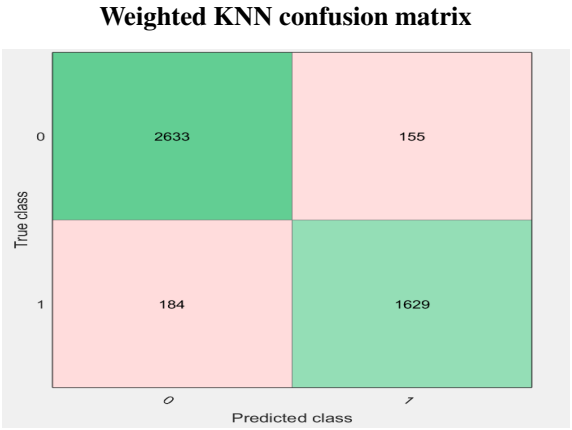

Figure 3. Weighted KNN confusion matrix.

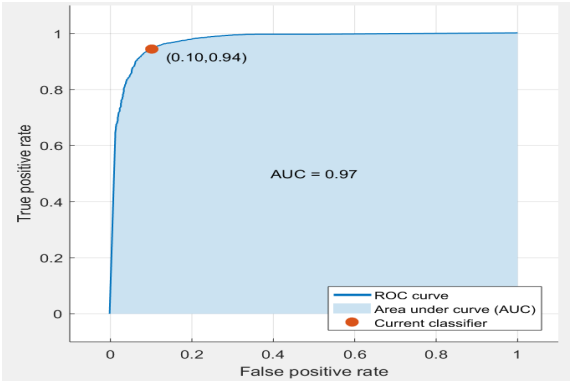

Figure 4. Weighted KNN ROC curve with a threshold of 0.5??

3. Advanced classifiers

3.1. Weighted k-nearest neighbors

The k-nearest neighbors (KNN) algorithm classifies the input data by finding the k nearest training points to the point that is being classified. The prediction is the class that has the majority of the nearest neighbors as part of that class. To decide which of the k instances in the training set are closest to the new input, an Euclidean distance measure is used. Euclidean distance is calculated as the square root of the sum of the squared differences between a new point and an existing point across all input features.

For k=10 (considering the 10 closest training points) and weighting each neighbor by $\frac{1}{d^2}$, where d is the distance between this point and the point being classified, the model achieves an accuracy of 92.6%. 10-fold cross validation was again implemented to reduce over fitting.

The overall accuracy (in this case) is almost equivalent to that of logistic regression, as is the false positive rate.

3.2. Support vector machines

Support vector machines (SVM) is one of the most successful classification algorithms, it is more complex than logistic regression but usually offers better results. SVM give a set of weights, one for each feature, whose linear combination predicts the output. SVM maximize the margin around the hyperplane that separates the different data classifications. Maximizing the margin reduces the number of weights that are nonzero to the ones that correspond to important features in deciding the separating hyperplane (i.e the points which influence optimality are the ones that are close to the decision boundary). The Large margin condition excludes dichotomies.

The Linear SVM model achieved an accuracy of 92.6% (worse than logistic regression 92.7%). The false positive rate (predicting non-Spam as Spam) was 5%. The confusion matrix is shown in figure 5. The main disadvantage of this model is that having a false positive rate that is practically 0% correctly classifies Spam approximately 50% of the time shown by the ROC curve in figure 6.

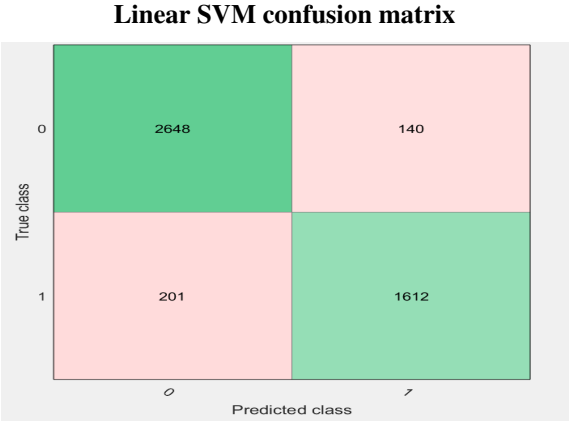

Figure 5. Linear SVM confusion matrix.

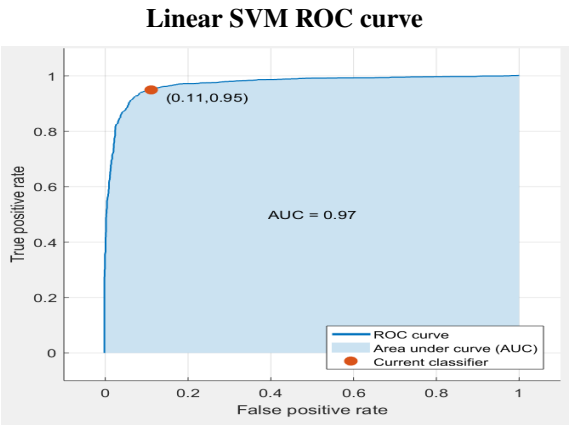

Figure 6. Linear SVM ROC curve

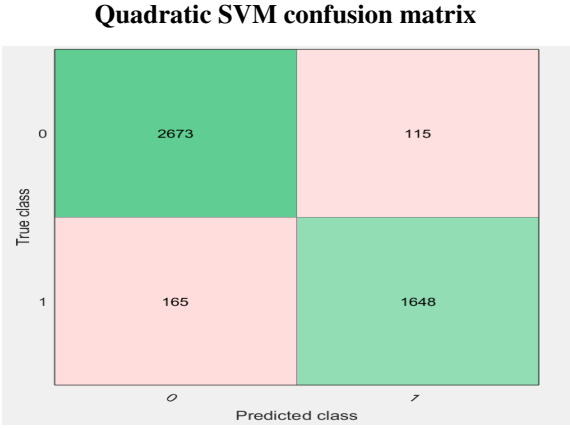

Figure 7. Quadratic SVM confusion matrix.

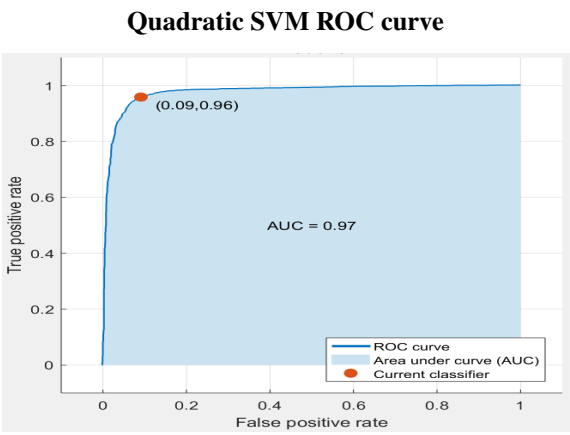

Figure 8. Quadratic SVM ROC curve

The quadratic SVM model out preforms all previous models with an accuracy of 93.9%, predicting Spam correctly 91% of the time and only incorrectly predicting non-Spam 4% of the time. Figure 7 shows the confusion matrix for the quadratic SVM. The quadratic SVM retains the advantage of Linear SVM, correctly classifying Spam 45% of the time while having a false positive rate of approximately 0%, this is shown in the ROC curve in figure 8.

4. Performance comparisons

4.1. Comparing advanced and baseline classifiers

Remarkably none of the more advanced algorithms significantly outperformed logistic regression, the key results are summarized in table 1. As stated earlier the quadratic SVM model achieved the highest overall accuracy of 93.9%, but this is not significantly better than logistic regression (92.7%) or even the linear SVM model (92.6%).

The quadratic SVM model obtained the lowest false positive rate (incorrectly predicting non-Spam as Spam) of 4%, while all other models had false positive rates of 5%. However if a condition is set such that the false positive rate must be approximately 0%, then logistic regression outperforms all models with a true positive rate of almost 60% closely followed by liner SVM (45-50%).

The main disadvantage of the KNN algorithm is that there is a large complexity associated with finding the nearest neighbors to a point. This is shown by the relatively slow prediction speed of approximately 1400 obs/sec compared to all the other models which have prediction speeds in the range of 15000 obs/sec (linear SVM) to 21000 obs/sec (quadratic SVM).

4.2. Proposed solution

From the summarized results in table 1, the proposed solution to predicting Spam emails would be logistic regression. This is because classifying non-Spam as Spam is highly undesirable and so a condition should be set to ensure an almost zero false positive rate. Under this condition logistic regression outperforms the other models with a true

positive rate of almost 60%. Logistic regression is also the simplest model (which is remarkable as it out preforms the more advanced models). Overall logistic regression is simple, quick to train, and provides the best results under the required conditions.

5. Fine tuning logistic regression

5.1. Principle component analysis

Principle component analysis (PCA) is used to reduce the number of features, (reduces the dimensionality of the training data set). It does this by removing features that are sufficiently correlated and transforming the data set so that the remaining features (principle components) are orthogonal.

When PCA is used to explain 95% of the variance only 2 of the 57 features are kept. This model has an overall accuracy of 72.2% with a false positive rate of 8%. Although the performance of this model is poor compared to the previous models it only uses 2 features compared to the previous 57 features that were used. This proves that many of the features have a high correlation as so various features will be redundant.

Using PCA to explain 99.999% of the variance results in 21 of the 57 features being used. The overall accuracy for this model is 90.9% with a training time of only 3.64 seconds. The confusion matrix for this model is show in figure 9 and the ROC curve is illustrated in figure 10. As shown by the ROC curve a true positive rate of $\approx 55\%$ can be achieved while maintaining a false positive rate of approximately 0%.

5.2. Conclusion

Overall a simple Spam email filter can be constructed to filter out $\approx 55\%$ of Spam emails while misclassifying $\approx 0\%$ of non-Spam emails. The Spam filter can be constructed using logistic regression, and with PCA only 21 of the 57 features need to be used. This simple model surprisingly outperforms the more advance models trailed here.

However this simple but effective Spam filter is by no means the best solution. In 2015 via the use of deep neural network models used with TensorFlow, Google's Spam filter had achieved a 99.9% accuracy with a false positive rate of 0.05% [1].

Logistic regression with PCA confusion matrix

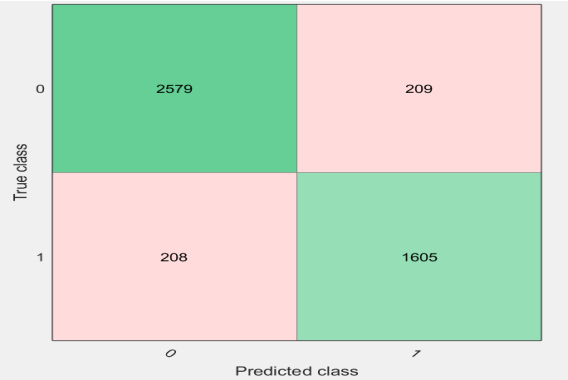

Figure 9. Confusion matrix for logistic regression with PCA explaining 99.999% of the variance.

Logistic regression with PCA ROC curve

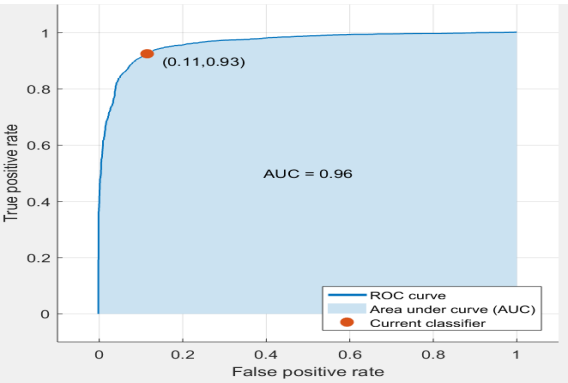

Figure 10. ROC curve with a threshold of 0.5?? for logistic regression with PCA explaining 99.999% of the variance.

6. References

1. Google statistics
https://blog.google/products/g-suite/how-machine-learning-g-suite-makes-people-more-pro
https://www.wired.com/2015/07/google-says-ai-catches-99-9-percent-gmail-spam/

Table 1. Comparison of classification models

	Overall accuracy (%)	False positive rate (%)	Largest true positive rate when false positive rate $\approx 0\%$ (%)	Training time (sec)	Prediction Speed (obs/sec)
Logistic regression	92.7	5	≈ 60	10.0	18000
Weighted KNN	92.6	5	≈ 30	5.81	15000
Linear SVM	92.6	5	≈ 50	12.4	1400
Quadratic SVM	93.9	4	≈ 45	12.8	21000