Assignment 1
Experiments with spam classification

T-710-MLCS, Machine Learning in Cyber Security

Reykjavik University - School of Computer Science, Menntavegi 1, IS-101 Reykjavík, Iceland

Damiano Pasquini
`damiano23@ru.is`

05. September 2023

1 Introduction

The goal of this assignment was to compare three supervised learning approaches used in spam detection: the Naive Bayes classifier, the Key Nearest Neighbors classifier and the Logistic Regression classifier. Initially, the main task is to pre-process data in order to create the dataset with train and test data. This is done by deleting for example the word "Subject:" from the email, removing non-alpha-numeric characters and single-character words. For each classifier is asked to obtain the following metrics: accuracy, precision, recall, F1-score, and the area under the ROC curve using the available Python libraries, such as *sklearn* and *nltk*. The entire code is developed in a Jupyter Notebook; here there are four functions to better divide the functionalities as pre-processing and model training.

The metrics used to extrapolate a comparison between these models are:

1. Accuracy: how often a model correctly predicts the outcome, expressed as the ratio of correct predictions to the total number of predictions made. It's a common metric used to evaluate the performance of classification models.

2. Precision: is a measure of the accuracy of positive predictions, indicating the proportion of correctly predicted positive instances among all instances predicted as positive.

3. Recall: measures the ability of a model to correctly identify all relevant instances within a dataset, minimizing false negatives. It is the ratio of true positive predictions to the total number of actual positive instances.

4. F1-score: that combines precision and recall to measure the accuracy of a binary classification model. It balances the trade-off between correctly identifying positive cases (recall) and minimizing false positives (precision).

5. AUC: (Area Under the Receiver Operating Characteristic Curve) is used to measure the performance of a binary classification model. It quantifies the model's ability to distinguish between positive and negative classes across various thresholds, with a higher AUC indicating better discrimination.

For each of the following models, there is a graph representing the ROC curves and a table with the aforementioned metrics.

2 Results

2.1 Influence of the dictionary size on Naive Bayes Classifier

In this implementation, as well as in the other classifiers, is used CountVectorizer to create a sparse matrix of word count vectors.

This vectorizer is sequentially used to create X_train and X_test which are respectively the mails (pre-processed) in the train set and the mails in the test set. After that, the classifier is instantiated and with the *fit* method, the model is trained.

The Naive Bayes classifier is trained using the following values assigned to the dictionary size: 100, 500, 1000, 2000, 3000. After having trained the model, we are going to obtain all the metrics. The dictionary size that gives a better accuracy is 3000 as we can see above.

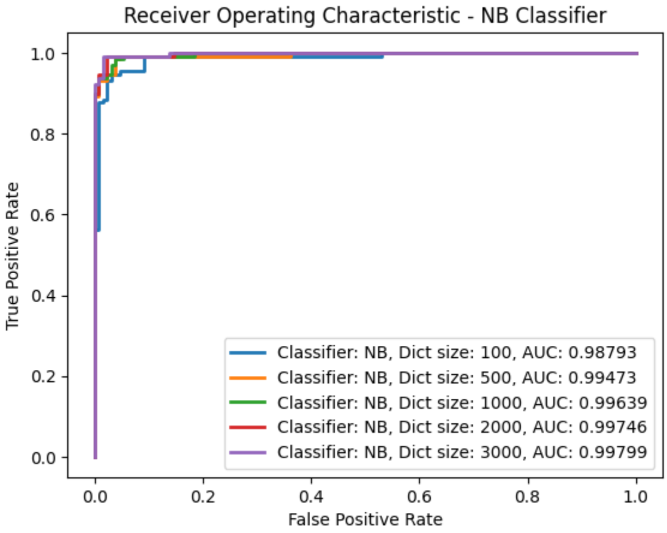

Figure 1: Output of the ROC curve referred to Naive Bayes Classifier with different values of Dictionary Size, proving the best performance with 3000.

	Dictionary size				
Metrics	100	500	1000	2000	3000
Accuracy	0.95	0.946154	0.946154	0.957692	0.957692
Precision	0.97561	0.991525	0.991525	0.991736	0.991736
Recall	0.923077	0.9	0.9	0.923077	0.923077
F1-score	0.948617	0.943548	0.943548	0.956175	0.956175
AUC	0.98793	0.99473	0.99639	0.99746	0.9979

Table 1: Metrics referred to Naive Bayes Classifier with different parameters of dictionary size

2.2 Influence of the K-parameter on Key Nearest Neighbors classifier

This classifier is implemented following the same structure as the others in this report. The only parameter that changes is the K-parameter (called k_param in the code); important to notice is the dictionary size (dict_size) that is fixed to 3000 considering the previous results from the Naive Bayes Classifier.

From the metrics of this training, the best result is given by the K-parameter set to 20.

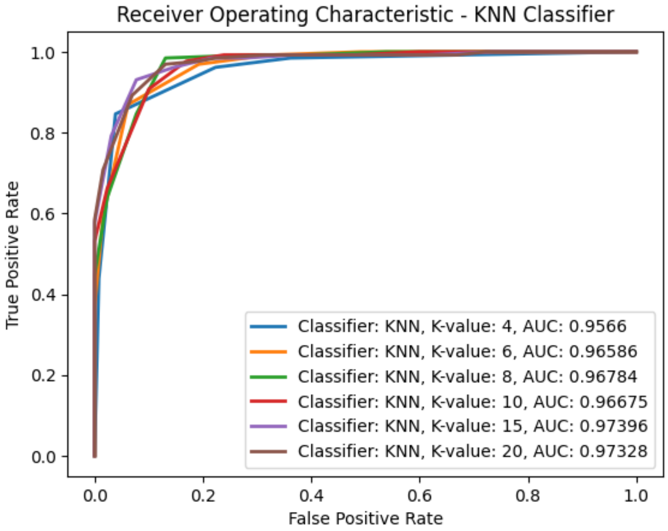

Figure 2: Output of the ROC curve referred to KNN Classifier with different values of K-parameter, proving the best performance with 20.

	K-value					
Metrics	4	6	8	10	15	20
Accuracy	0.903846	0.903846	0.884615	0.903846	0.907692	0.919231
Precision	0.956522	0.933884	0.916667	0.900763	0.868056	0.881119
Recall	0.846154	0.869231	0.846154	0.907692	0.961538	0.969231
F1-score	0.897959	0.900398	0.88	0.904215	0.912409	0.923077
AUC	0.9566	0.96586	0.96784	0.96675	0.97396	0.97328

Table 2: Metrics referred to KNN Classifier with different values of K-parameter

2.3 Influence of hyper-parameters on Logistic Regression

This part of the assignment asked to compare the Logistic Regression classifier using different values of C-parameter. Those values are 0.01, 0.1, 1, 10, 100. The implementation remains the same as the previous ones, but the C-parameter is added as an input in the function created in Python. After having evaluated the metrics, it results that better accuracy is gained by setting C-parameter at 100.

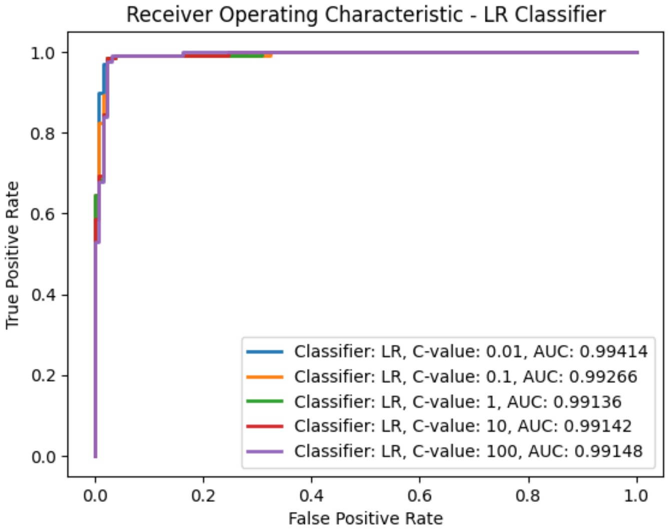

Figure 3: Output of the ROC curve referred to Logistic Regression Classifier with different values of C-parameter, proving the best performance with 100.

	C-parameter				
Metrics	0.01	0.1	1	10	100
Accuracy	0.973077	0.973077	0.973077	0.973077	0.976923
Precision	0.984252	0.962406	0.962406	0.962406	0.969697
Recall	0.961538	0.984615	0.984615	0.984615	0.984615
F1-score	0.972763	0.973384	0.973384	0.973384	0.977099
AUC	0.99414	0.99266	0.99136	0.99142	0.99148

Table 3: Metrics obtained from Logistic Regression classifier with different values of C-parameters.

2.4 Influence of hyper-parameters on Logistic Regression (II)

This task asked to compare the Logistic Regression Classifier, once trained with train data and once with test data, by the two metrics of Precision and Recall.

	Test		Train	
C-val	Precision	Recall	Precision	Recall
0.01	0.984252	0.961538	1	0.982906
0.1	0.962406	0.984615	1	1
1	0.962406	0.984615	1	1
10	0.962406	0.984615	1	1
100	0.969697	0.984615	1	1

Table 4: Precision and recall metrics obtained from Logistic Regression classifier with different values of C-parameters, applied on both train and test sets.

2.5 Comparison between different approaches

The following ROC curves indicate that the classifier that better classifies spam and not spam email is the Naive Bayes Classifier trained with a dictionary of 3000 words, with an Area-Under-the-Curve index of 0.99799.

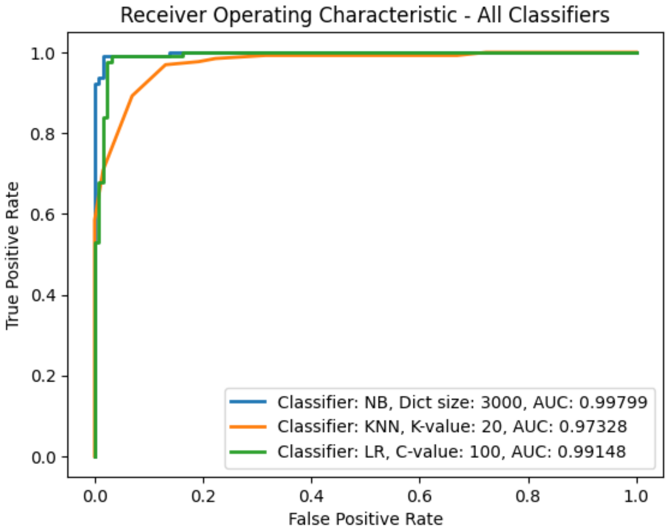

Figure 4: Output of the Receiver Operating Characteristic (ROC) curve of the three analyzed spam classifiers.