Spam Detection using Machine Learning

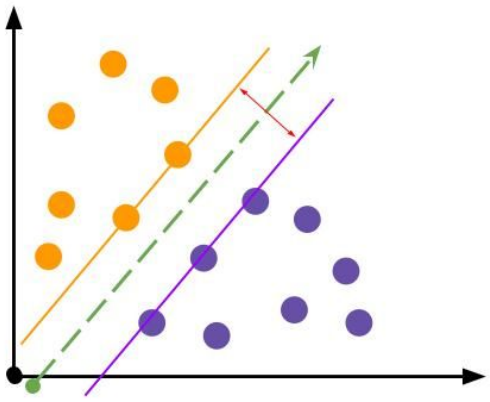
Group 6: January Shen, Ajinkya Sheth

Introduction:

Detection of spam in the email is the most popular application of machine learning (ML) algorithms. In this report, we present spam detection using two ML algorithms: Naive Bayes (NB) and Support Vector Machines (SVM). We use the dataset from https://github.com/savanpatel/machine-learning-101 repository which is divided into 702 training samples and 260 testing samples and implement our model using the sci-kit learn package in python. In this report, we start with a brief eexplanationof both the algorithms, lay out our machine learning process and finally compare the performance.

Support Vector Machines (SVM):

SVM generates an optimal hyperplane based on the labeled training set to classify new samples. Suppose we have a feature space of two variables and data is scattered across as shown in the diagram:



SVM algorithm will select an optimal line (green line) classifying the data into two regions. This representation is for two-dimensional space. For n-dimensional space, SVM will compute an optimal hyperplane. Note that for the above example itself there could multiple solutions or lines classifying the data, however, SVM will select the optimal one such that the margin (discussed below) is maximized.

Support Vectors: A key component of SVM are the support vectors (SV). SV's are those set of feature points which define the margin. They are called support vectors because they 'support' the classification in a way. In the figure above, SV's are denoted by Orange and Purple lines respectively.

There are four tuning parameters for SVM:
Kernel: It is a function of the input and the support vectors which helps in computing the hyperplane. A kernel could be linear, radial or exponential. It is based on some linear algebra.

Margin: (represented by the red line) It denotes the separation between the classes.

Regularization: It depicts the degree of accuracy of the model. It is also called the 'C parameter in the sci-kit library. Larger value C means our model will choose a smaller margin for classifying and a smaller C means the margin will be comparatively larger.

Gamma: It depicts how far the influence of a single training example reaches. In the figure above, on selecting a low value for gamma the optimum (green line) will be computed based on the nearby points only (closest ones are the points on the support vector). A higher value of gamma will result in consideration of the points farther away.

Naive Bayes Classification (NB):

We chose Naive Bayesian (NB) as another of our models because NB's rationale is intuitive and it usually works well in real practice albeit that the model is simple. A key point of NB is that it assumes all features are independent of each other. It is also a drawback because, in the real world, all the incidences may be related to each other. However, in practice, NB is still a powerful algorithm and works well in classification problems.

Naive Bayes model is based on the Bayes Theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

It tells us how often does A happen given that B has happened and given that we know the individual probabilities of occurrences of A and B and probability of B given A.

NB Algorithm calculates the probabilities of every feature, in our case, words belonging to a class 'Spam' or 'Ham' and selects the outcome with the highest probability.

NB utilizes the frequencies of words that show up in spams and creates a table with words and probabilities. When using the NB algorithm to determine whether an email is a spam, the algorithm calculates the frequencies of the words in the emails from the training model, finds out their corresponding probabilities of being spam, and calculate the possibility that the test data is spam based on the words that the target email contains.

Algorithm:

We divide our machine learning and evaluation process into four phases:

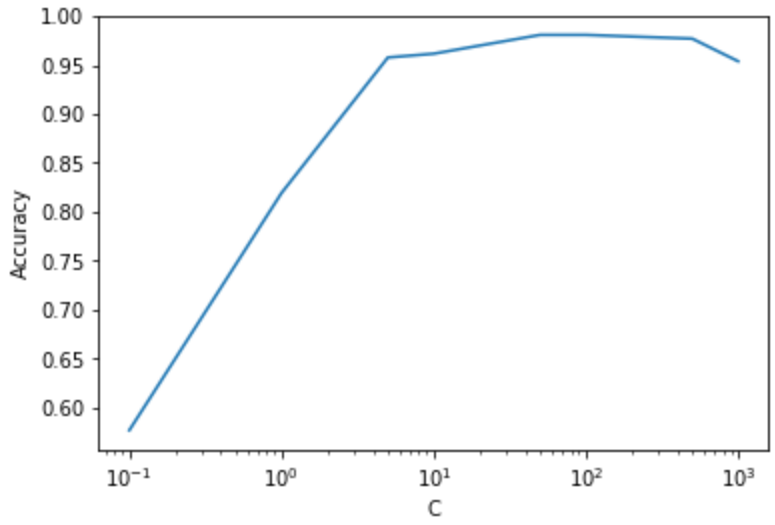
1. ETL and Data Cleaning: We load the data and create a feature matrix out of the words. For classifying spam, we do not need to consider all the text. Hence, we subject multiple filtering criteria over the text before selecting the words. We filter out numerical data, stopwords, and character with sequence length 1. Stopwords are those words which are neutral in nature and do not necessarily indicate whether the document is 'Spam' or 'Ham'. Common pronouns, verbs, and conjunctions can be considered as stopwords. To filter out the stopwords, we are using nltk package of python's Natural Language Processing repository.

   Moreover, out of the words selected we limit to 3000 most frequent words for model fitting. Thus, the feature matrix represents a matrix of m x n dimensions where m represents the number of emails (number of documents) and n represents the number of words (3000 in our case). For our training set, we have 702 samples, hence m=702 for training.

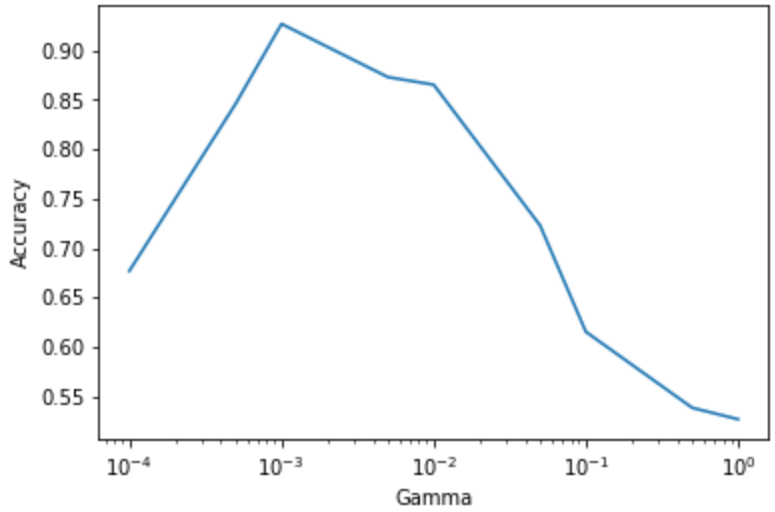
2. Model Fitting: In this phase, sci-kit learn's packages are used to train and test the model and compute its accuracy.
   a. SVM: From our initial training, it is evident that the accuracy metric of about 50% is not an indicator of good performance and our model needs tuning. We use the RBF (Radial Basis Function) kernel is this case as our data is non-linear in nature and RBF performs well with non-linear data.
   b. NB: It is a very simple ML model to implement. We use the Gaussian NB model as our feature matrix consists of the number of words which is a continuous variable and Gaussian NB model works well with continuous models.

3. Tuning and Optimization:
   a. SVMs require tuning for better performance and tuning is performed by varying C and gamma values. We can improve our accuracy (82%) of the untuned model by carefully selecting the parameters.

      First, we tune our model to the C-parameter. We notice that the model performs well for C=10 giving out an accuracy of more than 95%. For higher values of C, the accuracy remains almost the same and we chose C=10. From our discussion earlier, larger C value means the margin our model is comparatively smaller which makes sense as during spam classification there are a lot of words in the grey zone.



For gamma, we notice the model performs exceptionally well for gamma=0.001. Since this is a high dimensional feature space (of 3000 features) it is more plausible to choose a low value for gamma. That's because in high dimensional space the relative distances between points are much higher due to the Curse of Dimensionality and a smaller value of gamma means the points further away from the margin will influence the optimal hyperplane. Overall we achieve an accuracy of 97.3% for SVM model.



b. For NB: NB is much easier to train and we achieve high accuracy of 96.2% by the simple implementation. Hence, we skip the tuning process for NB.

4. Performance Metrics: While accuracy is a good indicator of how the model is doing overall it is not sufficient to compare between two different models. In our report, we

chose additional metrics for comparison (discussed in the next section) to differentiate between the performance of SVM to that of NB.

Performance comparison

For our performance rubrics, we chose Confusion Matrix, ROC, AUC, Precision, Recall, and F1 as our evaluation metrics. They are the several most important evaluation metrics for checking any classification model's performance.

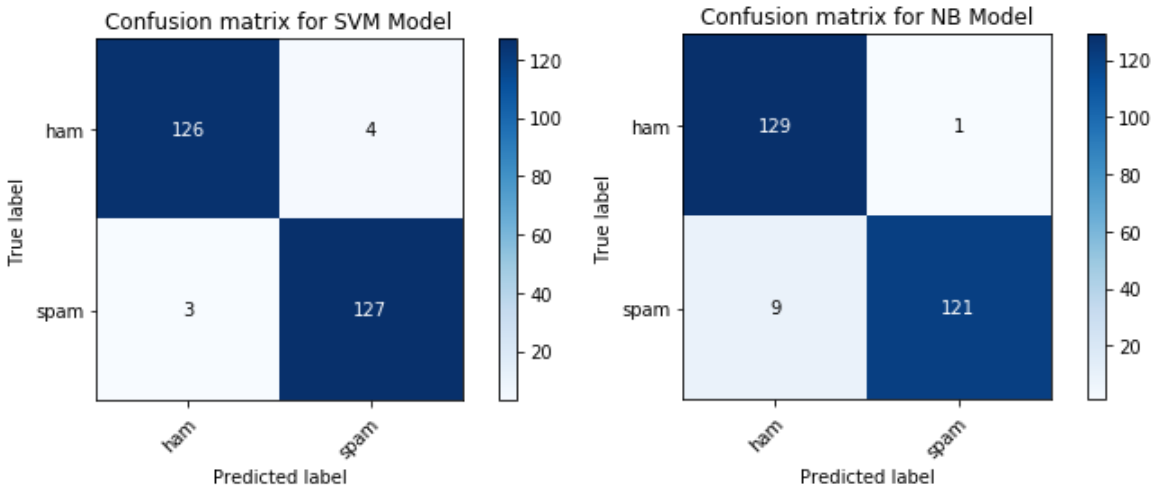
Confusion Matrix, Accuracy, Precision, Recall, and F1

Confusion Matrix describes the performance of a classification model. The two axes of the table are "predicted label" and "real label". It creates a 2x2 matrix where there are "True Positive", "True Negative", "False Positive", and "False Negative". The first word in the result represents whether the classification did it right or wrong, the second word in the result represents the decision that the classification made (Positive or Negative).

From Confusion Matrix, there stemmed some scoring criteria: Accuracy, Precision, Recall, and F1. Precision:

- Accuracy - a ratio of correctly predicted observation to the total observations. Accuracy = TP+TN/all.
- Precision - the ratio of correctly predicted positive observations to the total predicted positive observations. Precision = TP/(TP+FP).
- Recall (Sensitivity) - the ratio of correctly predicted positive observations to all observations in the actual class. Recall = TP/(TP+FN)
- F1 score - the weighted average of Precision and Recall. This is to avoid untrue representation of Precision and Recall when the data label is unbalanced (dataset severely skewed to on side of True or False). F1 = 2*(Precision*Recall)/(Precision + Recall)

The two matrices below are the Confusion matrices from our two algorithms. The left one is from SVM model, and the right one is from the NB model. The upper left corner is "True Positive". With the data size of 260, SVM correctly labeled 126 (TP) + 127 (FN) data and missed 4(FN) + 3 (FP) data, while NB correctly labeled 129 (TP) + 121 (FN) data and missed 9 (FN) + 1 (TP) data.

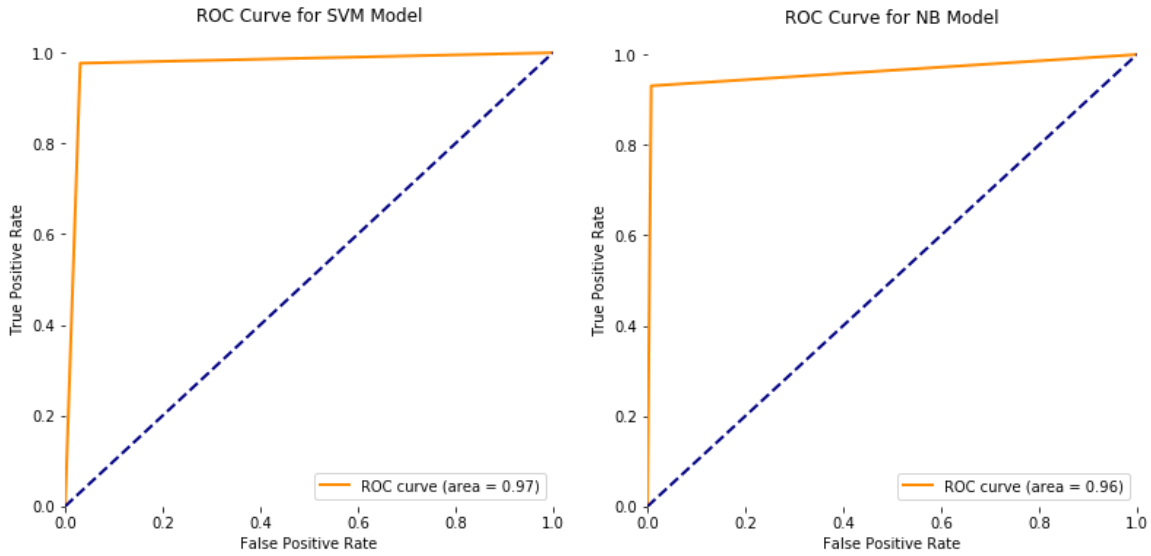


Overall, SVM performs better than NB in this dataset in terms of accuracy and precision. However, NB performs better eventually judging from f1 score because when the data has a positive label (ham), NB does better in labeling them as positive. This feature is important because we can thus have higher confidence in the predicted results that are labeled as positive are real positive data.

	SVM	NB
accuracy	0.973	0.962
recall	0.498	0.516
precision	0.977	0.935
f1	0.660	0.665

ROC and AUC

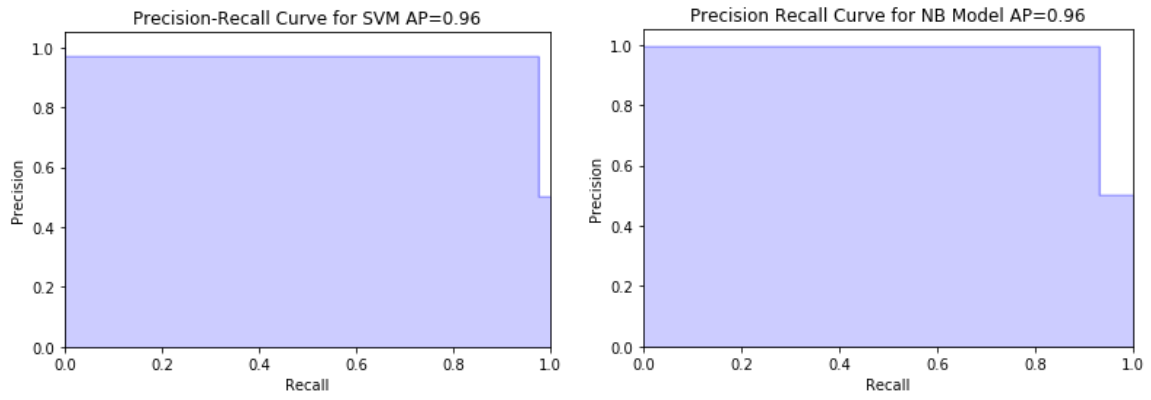
ROC (Receiver Operating Characteristics) and AUC (Area Under The Curve) is one of the most important evaluation metrics for checking a classification model's performance. ROC is a probability curve with the x-axis representing the FP rate and y-axis representing the TP rate. The curve represents the tradeoff of FP and TP rate when we give different determining threshold to the data (for example, label positive when it's >0.6, or label positive when it's >0.8); AUC is the area under ROC curve. The bigger the AUC, the better the model classifies the data.



The above left-hand side chart is the ROC curve for SVM model, and the right-hand side is the ROC curve for NB model. From bare eyes, the ROC curve looks not much different. In fact, the AUC is 0.97 for SVM and 0.96 for NB. Not a big difference, but SVM performs better in this model selection criterion.

Precision-Recall Curve

A precision-recall curve shows the relationship between precision and recall for every possible cut-off. The x-axis shows recall, which is TP / (TP + FN), and the y-axis shows TP / (TP + FP). Precision-Recall functions similarly to f1 score, the curve shows the tradeoff of precision and recall when we select different numbers for decision threashold. It is to prevent a misleading chart when the data is heavily skewed. From the below charts, the left-hand side is the precision-recall curve for SVM, and the right-hand side is the precision-recall curve for NB. We see that the pattern of the two models is similar.



References:

Patel, S. (2018, November 10). Chapter 2 : SVM (Support Vector Machine) — Theory. Retrieved May 22, 2019, from https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72

scikit-learn.org. (n.d.). Documentation scikit-learn: machine learning in Python — scikit-learn 0.21.1 documentation. Retrieved May 22, 2019, from https://scikit-learn.org/stable/documentation.html

Wikipedia. (2019, May 19). Naive Bayes Classifier. Retrieved May 22, 2019, from https://en.wikipedia.org/wiki/Naive_Bayes_classifier