# CS5691: Pattern Recognition and Machine Learning

Assignment 3

## Report

**Course Instructor**: Arun Rajkumar

**Submitted By**: Vennapareddy Abhigna

**Roll Number**: ME19B059

**Date:** 27/04/2022


Indian Institute of Technology Madras

Chennai 600036, India

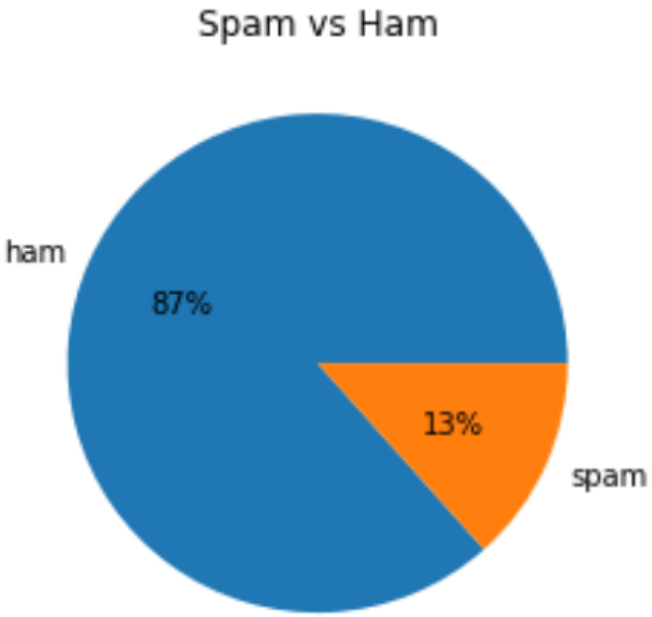
# SPAM or HAM

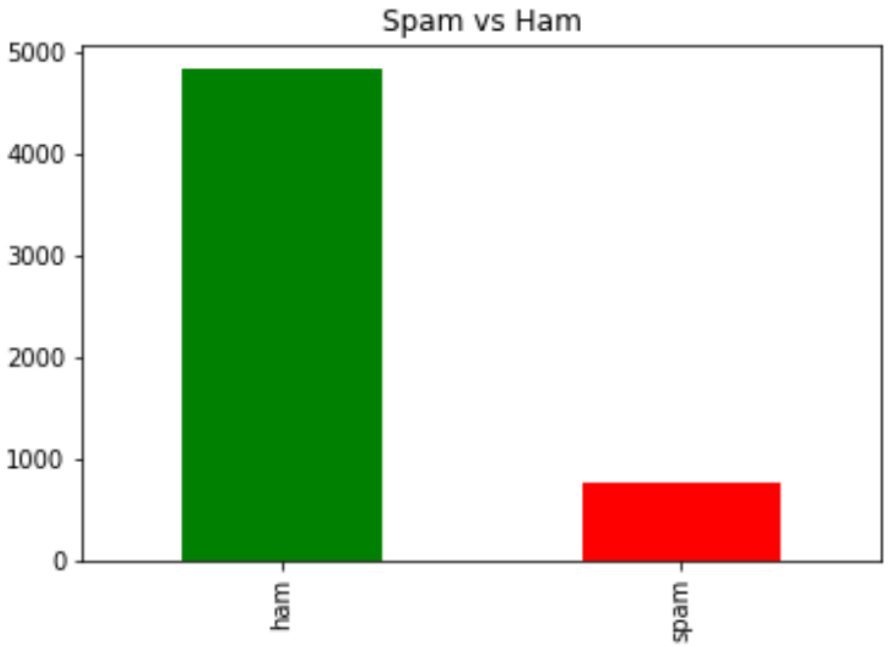
We built a spam classifier by training the data taken from Kaggle.

**Dataset:**

The dataset has been chosen from
https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset?resource=download .

This dataset consists of 5572 labeled messages with 87% being labeled as ham (0) and 13% labeled as spam (1).



### Reading files from test folder:

Give the path directory of the test folder in the second cell of the code file. Each file in this folder will be a test email and will be named 'email#.txt' ('email1.txt', 'email2.txt', etc).

The contents in these files are converted from rtf to text files and then stored in a dataframe. For each of these emails, the classifier predicts +1 (spam) or 0 (non Spam)

### Problem Analysis with dataset:

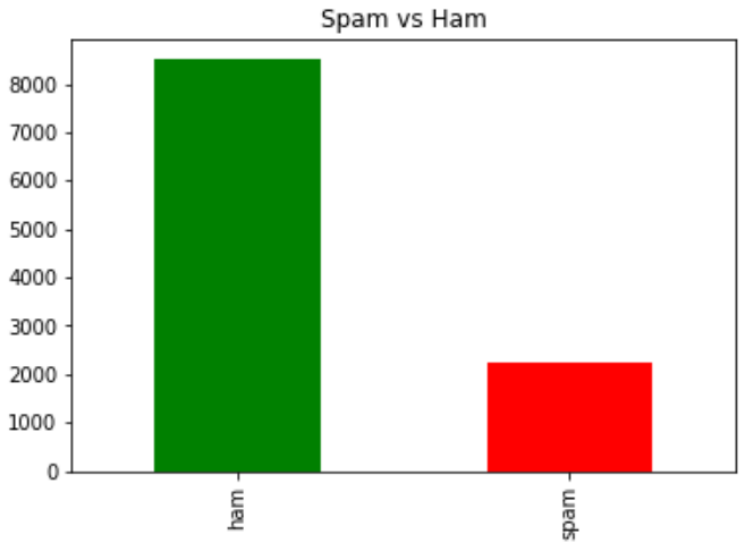
The below observed results prove that the dataset that we chose is not enough to give best possible results.

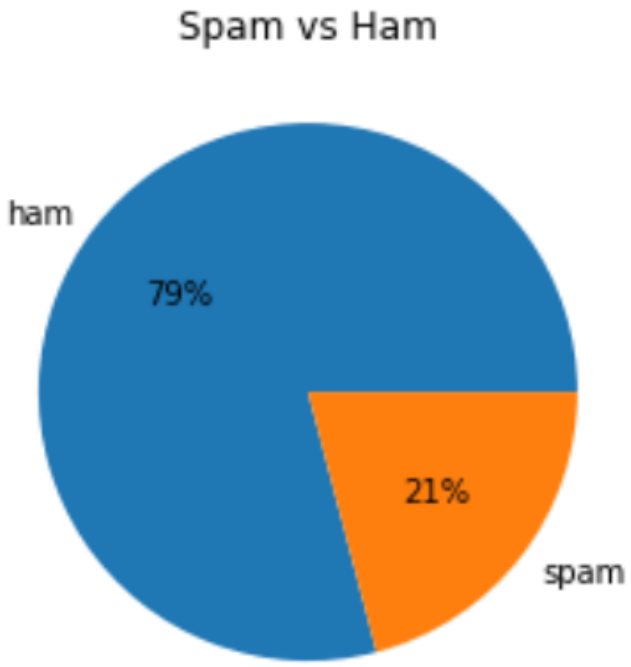
| | C | Train_Accuracy | Test_Accuracy |
|---|---|---|---|
| **0** | 5.0 | 0.999551 | 0.98296 |
| **1** | 10.0 | 0.999776 | 0.98296 |
| **2** | 15.0 | 1.000000 | 0.98296 |
| **3** | 20.0 | 1.000000 | 0.98296 |
| **4** | 25.0 | 1.000000 | 0.98296 |

Hence we add another dataset to this to get more training data. Dataset can be downloaded from: https://drive.google.com/file/d/1Ig0NFcTD4PbUKyL-8ZT9UXgWYwSO6pFg/view?usp=sharing

Now, by adding a new dataset we revisit all the results.

This final dataset consists of 10743 labeled messages with 79% being labeled as ham (0) and 21% labeled as spam (1)






## Vectorizing:

We use the CountVectorizer function as it is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation.

It produced 54003 features by mentioning the presence of each feature in each message by 0 or 1. These are then saved as a dataframe to train the model.

## Training and Testing using SVM Models:

The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space making it separable. Compared the results of kernel SVM with gaussian and sigmoid functions by splitting the data into 75% train and 25% test set.

Gaussian: $K(x_i, x_j) = exp(-\gamma||x_i - x_j||)^2$

Sigmoid: $K(x_i, x_j) = tanh(\alpha x^a y + c)$

In gaussian kernel, C for which the best test accuracy is obtained is chosen and predicted on the test emails.

| | C | Train_Accuracy | Test_Accuracy |
|---|---|---|---|
| **0** | 10.0 | 0.991932 | 0.972822 |
| **1** | 20.0 | 0.995159 | 0.971705 |
| **2** | 30.0 | 0.996525 | 0.971705 |
| **3** | 40.0 | 0.997766 | 0.970588 |
| **4** | 50.0 | 0.998759 | 0.970961 |
| **5** | 60.0 | 0.998883 | 0.969844 |
| **6** | 70.0 | 0.999255 | 0.968354 |
| **7** | 80.0 | 0.999379 | 0.967238 |
| **8** | 90.0 | 0.999379 | 0.967982 |
| **9** | 100.0 | 0.999628 | 0.967982 |
| **10** | 110.0 | 0.999752 | 0.967982 |
| **11** | 120.0 | 0.999752 | 0.967610 |
| **12** | 130.0 | 0.999752 | 0.966121 |
| **13** | 140.0 | 0.999752 | 0.965748 |
| **14** | 150.0 | 0.999876 | 0.965748 |
| **15** | 160.0 | 0.999876 | 0.966121 |
| **16** | 170.0 | 0.999876 | 0.966121 |
| **17** | 180.0 | 0.999876 | 0.966493 |
| **18** | 190.0 | 0.999876 | 0.966121 |

Confusion matrix for C = 10

| | Predicted 0 | Predicted 1 |
|---|---|---|
| **Actual 0** | 2121 | 19 |
| **Actual 1** | 54 | 492 |

In sigmoid kernel, C for which the best test accuracy is obtained is chosen and predicted on the test emails.

| | C | Train_Accuracy | Test_Accuracy |
|---|---|---|---|
| **0** | 10.0 | 0.947127 | 0.928891 |
| **1** | 20.0 | 0.945017 | 0.925168 |
| **2** | 30.0 | 0.966737 | 0.932614 |
| **3** | 40.0 | 0.939928 | 0.913999 |
| **4** | 50.0 | 0.937818 | 0.913254 |
| **5** | 60.0 | 0.935956 | 0.910648 |
| **6** | 70.0 | 0.935336 | 0.909531 |
| **7** | 80.0 | 0.934839 | 0.908414 |
| **8** | 90.0 | 0.934219 | 0.906925 |
| **9** | 100.0 | 0.933598 | 0.905808 |
| **10** | 110.0 | 0.933226 | 0.903574 |
| **11** | 120.0 | 0.933102 | 0.902829 |
| **12** | 130.0 | 0.932605 | 0.902085 |
| **13** | 140.0 | 0.932481 | 0.900596 |
| **14** | 150.0 | 0.932357 | 0.900596 |
| **15** | 160.0 | 0.931985 | 0.899851 |
| **16** | 170.0 | 0.931736 | 0.900596 |
| **17** | 180.0 | 0.931240 | 0.897990 |
| **18** | 190.0 | 0.931364 | 0.899479 |

Confusion matrix for C = 30

| | Predicted 0 | Predicted 1 |
|---|---|---|
| **Actual 0** | 2051 | 89 |
| **Actual 1** | 92 | 454 |

Compared to both gaussian and sigmoid kernels, Gaussian kernel gave best results.

By both the functions, the given sample emails are labeled correctly as 0 and 1 respectively for email1.txt and email2.txt.