**Spam Email Detection - UCI Spam base Data Set**

Semester project report for Data Mining, Dr. Xiaofei Zhang instructing

Krishnaveni Mandadi, U00829899 Department of Computer Science, University of Memphis, Memphis, TN, [kmandadi@memphis.edu](mailto:kmandadi@memphis.edu)

**ABSTRACT**

Spam emails have become a genuine issue in recent years, and the amount of spam emails has increased as the number of internet users has increased. These spammers target individuals who are unaware of the frauds, and if we click by accident or on purpose, it can lead to the theft of confidential information, or getting the individual to submit crucial information into a phony website, crashing your system. As a result, it is vital to distinguish fraudulent spam letters; this project will accomplish this by employing machine learning techniques and evaluating all these algorithms on our data sets in order to determine the optimal algorithm for email spam detection with the maximum precision and accuracy.

1. **Introduction**

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are being use by the machine learning community for the empirical analysis of machine learning algorithms. The archive created as an ftp archive in 1987 by David Aha and fellow graduate students at UC Irvine. Since that time, it has been widely using by students, educators, and researchers all over the world as a primary source of machine learning data sets. As an indication of the impact of the archive, it has been cited over one thousand times, making it one of the top one hundred most cited "papers" in all of computer science. The current version of the web site was designed in 2007 by Arthur Asuncion and David Newman, and this project is in collaboration with [Rexa.info](http://rexa.info/) at the University of Massachusetts Amherst and funding support from the National Science Foundation is gratefully acknowledged.

* 1. **Dataset**

Our collection of spam e-mails came from our mail carrier and individuals who had filed spam. Our collection of non-spam e-mails came from filed work and personal e-mails, and hence the word 'george' and the area code '650' are indicators of non-spam. These are useful when constructing a personalized spam filter. One would either blind such non-spam indicators or get a very wide collection of non-spam to generate a general-purpose spam filter.

The last column of 'spambase.data' denotes whether the e-mail was considered spam (1) or not (0), i.e., unsolicited commercial e-mail. Most of the attributes indicate whether a particular word or character was frequently occurring in the e-mail. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters. For the statistical measures of each attribute, see the end of this file. Here are the definitions of the attributes:

48 continuous real [0,100] attributes of type word\_freq\_WORD = percentage of words in the e-mail that match WORD,

i.e., 100 \* (number of times the WORD appears in the e-mail) /total number of words in e-mail.

A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.

6 continuous real [0,100] attributes of type char\_freq\_CHAR = percentage of characters in the e-mail that match CHAR,c

i.e., 100 \* (number of CHAR occurrences) / total characters in e-mail

1. **Problem Statement**

In supervised or inductive machine learning, the algorithms learn from the training dataset that contains both inputs and outputs (results) and a model created. The model is then evaluating for new samples for classification. In case of binary classification, the output belongs to two classes. In recent days e-mail spam filtering is one of the important research fields.

1. **Solving the Problems**

Over the period several machine learning techniques such as Linear Regression, SVM, Decision Tree techniques etc. have been used in classifying spam email datasets. All these techniques use different approaches to solve the problem, where it tries to model the data like human brain processing information.

* 1. **Dataset Exploration**

The Data set consists of nearly 4601 email files and data consists of either spam or hams. Spams, aka junk emails, are unsolicited messages sent in bulk by email. Hams are non-spams expected by email recipients. Amount the spam emails in the dataset are 1813 and non-spam are 2788 and data is read and described by the Python with various packages.

* 1. **Data Preprocessing**

Major tasks in data pre-processing are data cleaning, data integration, data transformation and data reduction.

* Remove if any duplicate row level records
* Check for any Nulls then remove the redundant data which do not give useful information in the process of building effective data model.
  + 1. **SMOTE techniques**

SMOTE (synthetic minority oversampling technique) is one of the most used oversampling methods to solve the imbalance problem. It aims to balance class distribution by randomly increasing minority class examples by replicating them.

After Dropping Duplicate records, Now dataset has become 1679 with spam (1) and 2531 ham(0) records.

* Apply the imblearn.over\_sampling technique to balance the data and after applying smote data became 2531 spam and 2531 ham records.
  + 1. **Standardization**

Standardizing a dataset involves rescaling the distribution of values so that the mean of observed values is 0 and the standard deviation is 1. It is useful, and even required when your data has input values with differing scales.

* I have applied standardscalar technique for standardization.
  1. **Feature Extraction**

Feature selection in supervised learning has a main goal of finding a feature subset that produces higher classification accuracy

* + 1. **Correlation**

Correlation plot between the variables convey lot of information about the relationship between them. But it is difficult to derive proper insights as the feature space is too large. We can try Feature Extraction Techniques to get lesser features, covering the variability.

* Applied correlation method to find the relationship among the features.
  + 1. **Linear Discrimination Analysis**

LDA is a classifier used to find a linear combination of features, which separates two or more classes of data. The succeeding combination can be used as a linear classifier. In LDA, the classes are expected to be normally distributed. LDA is based on classic estimators of location and covariance, and that is why the method is sensitive to outlying samples decreasing the LDA performance with an increase in the number of incorrectly assigned samples.

* Applied Linear Discrimination Analysis technique to reduce the dimensionality.

1. **Methodology**
   1. **Performance evaluation measures:**

Spam filters typically assess enormous databases of ham and spam messages that are available to the public. However, it has been observed that utilizing accuracy alone as a performance metric is insufficient. Other performance indicators, such as recall and precision, should be taken into account. When a spam message is incorrectly labeled as ham, it creates a little issue because all the user has to do is delete the message. When a non-spam communication is incorrectly tagged as Spam, on the other hand, it is annoying because it implies the risk of losing vital information due to the filter's classification error.

* 1. **Modeling with Different Algorithms**

Supervised machine learning techniques applied in the experiment to the Spam Email dataset. The performance of the techniques discussed below. Out of all five machine-learning techniques considered in the experiment, SVM performed with good accuracy since SVM performed well then I choose ensemble techniques considering SVM as base but Logistic Regression is simple and computationally fast.

* + 1. **Logistic Regression**

It is one of the appropriate algorithms used for classification of datasets. In case of classifying a dataset named as spam base the logistic regression is the most versatile decision-based approach for detecting spam emails out of a dataset.

Accuracy = 93.0%

F1 Score = 93.0%

**Confusiton Matrix:**

[[484 29]

[ 42 458]]

**Classification Report:**

precision recall f1-score support

0 0.92 0.94 0.93 513

1 0.94 0.92 0.93 500

accuracy 0.93 1013

macro avg 0.93 0.93 0.93 1013

weighted avg 0.93 0.93 0.93 1013

* + 1. **SVM**

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. The standard SVM takes a set of input data and predicts, for each given input, which of two classes comprises the input, making the SVM a non-probabilistic binary linear classifier.

Accuracy = 93.60000000000001%

F1 Score = 93.60000000000001%

**Confusiton Matrix:**

[[482 31]

[ 34 466]]

**Classification Report:**

precision recall f1-score support

0 0.93 0.94 0.94 513

1 0.94 0.93 0.93 500

accuracy 0.94 1013

macro avg 0.94 0.94 0.94 1013

weighted avg 0.94 0.94 0.94 1013

### **Ensemble classifiers**

[Ensemble learning](https://www.sciencedirect.com/topics/computer-science/ensemble-learning) is a new approach in which a group of different classifiers are trained and assembled to further improve the [classification accuracy](https://www.sciencedirect.com/topics/computer-science/classification-accuracy) of the complete system on identical problem, in this case it is spam filtering. They are a class of machine learning algorithm that work in harmony and are applied to enhance the classification performance of the entire system.

1. **AdaBoostClassifier**

Accuracy = 82.0%

F1 Score = 82.0%

**Confusiton Matrix:**

[[410 103]

[ 79 421]]

**Classification Report:**

precision recall f1-score support

0 0.84 0.80 0.82 513

1 0.80 0.84 0.82 500

accuracy 0.82 1013

macro avg 0.82 0.82 0.82 1013

weighted avg 0.82 0.82 0.82 1013

1. **Bagging**

Accuracy = 93.4%

F1 Score = 93.4%

**Confusiton Matrix:**

[[480 33]

[ 34 466]]

**Classification Report:**

precision recall f1-score support

0 0.93 0.94 0.93 513

1 0.93 0.93 0.93 500

accuracy 0.93 1013

macro avg 0.93 0.93 0.93 1013

weighted avg 0.93 0.93 0.93 1013

* + 1. **Gradient Boosting Classification**

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n\_classes\_ regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced.

Accuracy = 92.7%

F1 Score = 92.7%

**Confusiton Matrix:**

[[481 32]

[ 42 458]]

**Classification Report:**

precision recall f1-score support

0 0.92 0.94 0.93 513

1 0.93 0.92 0.93 500

accuracy 0.93 1013

macro avg 0.93 0.93 0.93 1013

weighted avg 0.93 0.93 0.93 1013

**Results Of Classification Algorithms**

|  | **Accuracy** | **Precision** | **Recall** | **F1-score** | **AUC-ROC score** |
| --- | --- | --- | --- | --- | --- |
| **Logistic Regression (LR)** | 93.1 | 94.2 | 91.6 | 92.9 | 93.1 |
| **Support Vector Machine (SVM)** | 93.5 | 93.9 | 92.8 | 93.4 | 93.5 |
| **Ensemble Boosting with SVM (SvmBoost)** | 77.2 | 71.7 | 88.8 | 79.4 | 77.3 |
| **Ensemble Bagging with SVM (SvmBag)** | 93.6 | 93.9 | 93.0 | 93.5 | 93.6 |
| **Gradient Boosting (GB)** | 93.1 | 93.5 | 92.4 | 93.0 | 93.1 |

1. **Observations**

* The Dataset was small around 4600 samples & after preprocessing 6.2% of the data samples dropped.
* The spam emails were 20% more than non-spam ones, hence SMOTE Technique applied on the data to balance the classes, adding 13.5% more samples to the dataset.
* The large feature set reduced by Feature Extraction Technique -LDA

1. **Conclusion**

From the obtained results, the ensemble & Gradient boosting algorithms perform the better on the current dataset, followed by Support Vector Machines and the bagging method has produced more than 93.7% accuracy in spam detection. For the current problem statement, it is more important to focus on the true positives, and is fine to misclassify a spam email, but not have a regular mail classified as spam. Hence, we shall focus on the Precision Score to evaluate the current model.

1. **Deliverables**

All the spam email project documents and code are uploaded at my github account.

<https://github.com/KrishnaveniReddyMandadi/SpamEmailProject>

**REFERENCES**

[1] “UCI Machine Learning spambase Data Set”. [UCI Machine Learning Repository: About](https://archive.ics.uci.edu/ml/about.html)

[2] “Comparative Study on Email Spam Classifier using Data Mining Techniques, R. Kishore Kumar, G. Poonkuzhali,P.Sudhakar, *Member, IAENG* “. [Comparative\_Study\_on\_Email\_Spam\_Classifier\_using\_D.pdf](file:///C:\Users\kmandadi\Downloads\Comparative_Study_on_Email_Spam_Classifier_using_D.pdf)

# [3] “Machine learning for email spam filtering: review, approaches and open research problems”.

# [Machine learning for email spam filtering: review, approaches and open research problems - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S2405844018353404) **[4]** “**Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges”** [(PDF) Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges | Hamza Aldabbas - Academia.edu](https://www.academia.edu/77280760/Machine_Learning_Techniques_for_Spam_Detection_in_Email_and_IoT_Platforms_Analysis_and_Research_Challenges) [5] “**Evaluation of Machine Learning Techniques for Email Spam Classification”** [(PDF) Evaluation of Machine Learning Techniques for Email Spam Classification (researchgate.net)](https://www.researchgate.net/publication/354383033_Evaluation_of_Machine_Learning_Techniques_for_Email_Spam_Classification)