Spam Ham Classifier

Revision Number: 1.1
Last date of revision: 29/09/2021

Document Version Control

Date Issued	Version	Description	Author
28/09/2021	1	Initial HLD - V1.0	Geetha S Akanksha Venkatesh V B Ashok Kumar S M C Shravan
29/09/2021	1.1	Final HLD - V.1.1	Geetha S Akanksha Venkatesh V B Ashok Kumar S M C Shravan

Contents

Abstract

The high number of unsolicited emails commonly called spam has necessitated the development of an increase in reliable and robust antispam filters. Using the recent machine learning approach we have been successful in detecting and filtering spam emails. We are classifying Spam or ham from the dataset which is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according to being ham (legitimate) or spam.

Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface is implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:

  Security
  Reliability
  Maintainability
  Portability
  Reusability
  Application compatibility
  Resource utilization
  Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	*Description*
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment

2. General Description

2.1 Product Perspective

The Spam Ham Classifier is a machine learning-based web app that will help us to classify whether the SMS is spam or ham.

2.2 Problem statement

- To classify Spam or ham from the dataset which is a set of SMS tagged messages that have been collected for SMS Spam research.
- It contains one set of SMS messages in English of 5,574 messages, tagged according to being ham (legitimate) or spam.

2.3 Proposed solution

The solution proposed here is a Spam Ham Classifier which will help to classify Spam or ham from the given dataset. Here we have used Logistic regression, Extra trees classifier, Adaboost classifier, XGBoost classifier and Gradient boosting classifier.

2.4 Further improvements

In Spam Ham Classifier by using rough set postprocessing approachable to significantly improve the accuracy. More data can be added to the datasets which help in getting high accuracy.

2.5 Data Requirements

Data requirements are completely dependent on the given problem statement.

- We need to use the dataset set which has been provided with both test and train data.
- The test and train datasets should have both input and output variables.
- The data sets should consist of the following:
  1. Label: spam, ham.
  2. Message: spam, ham.

2.6 Tools used

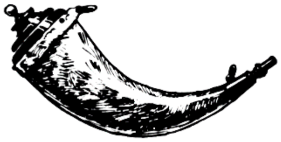
The tools used to build the model are Python programming language and frameworks such as NumPy, Pandas, Matplotlib and Seaborn, Scikit Learn, database.

This Photo by Unknown Author is licensed under CC BY-SA

This Photo by Unknown Author is licensed under CC BY-SA

This Photo by Unknown Author is licensed under CC BY-SA

This Photo by Unknown Author is licensed under CC BY-SA



2.6.1 Jupyter Notebook is used as IDE.

2.6.2 For visualization of the plots, Matplotlib, Seaborn are used.

2.6.3 Heroku is used for the deployment of the model.

2.6.4 Web application is created in Notepad++.

2.6.5 Front end development is done using HTML/CSS

2.6.6 Backend development is done using Flask

2.6.7 GitHub is used as a version control system.

2.6.8 MySQL Database is used

2.7 Hardware Requirements

- PC/ Laptop

2.8 Constraints

The Spam Ham Classifier is limited to the data provided in the datasets. Only the data provided in the dataset can be used for classifying spam and ham.

2.9 Assumptions

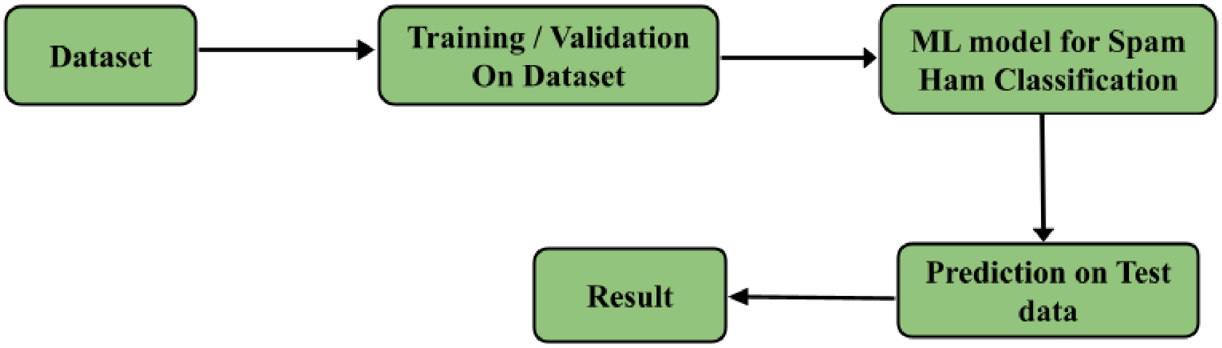
The main objective of the project is to implement the use cases as previously mentioned for an already existing dataset provided for doing the project. A machine learning-based algorithms named Logistic regression, Extra trees classifier, Adaboost classifier, XGBoost classifier and Gradient boosting classifier are used to classify spam and ham.

3 Design Details

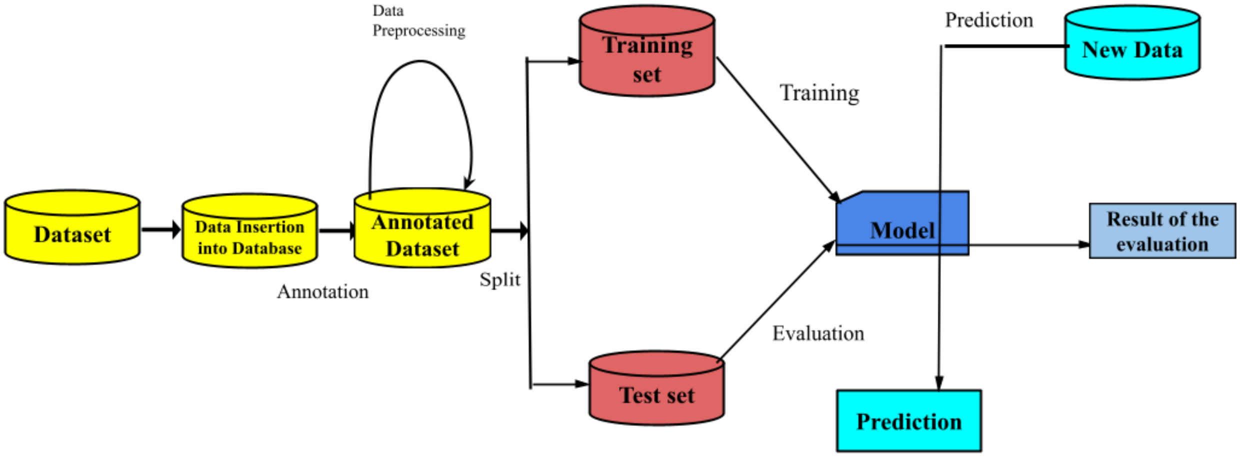
3.1 Process Flow

For the classification of spam and ham, we will use a machine learning model. The process flow diagram is as shown below.

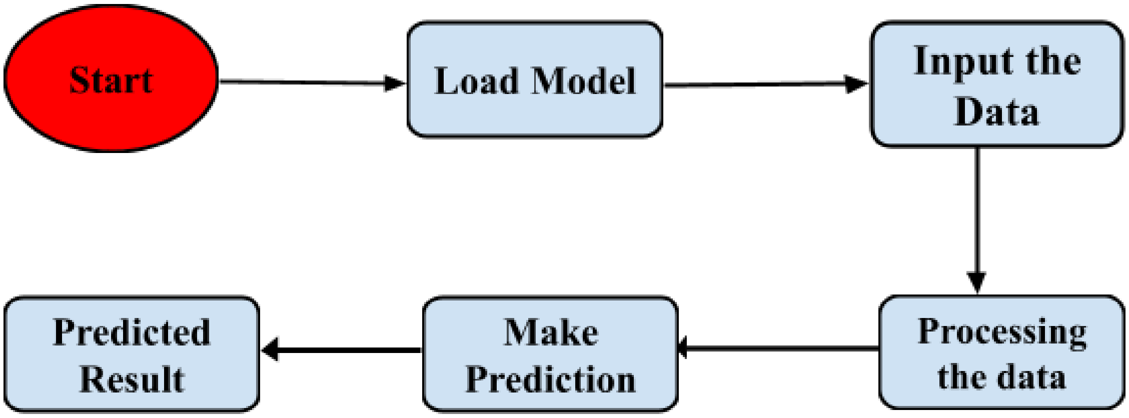
Proposed methodology


Dataset
Training / Validation On Dataset
ML model for Spam Ham Classification
Prediction on Test data
Result

3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 Event log

Every event is logged in the system so that the user can understand whichever process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging is required.
2. The System should be able to log each and every system flow.
3. The developer can choose the logging method. You can choose database logging/ File logging as well.
4. The system should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 Error Handling

When it comes to error handling, there are most possibly no errors because all the errors are handled during the execution of the program.

4 Performance

The Spam Ham Classifier is used for the classification of spam and ham based on the given dataset. In the designed web app, the input is given and the output is displayed within a fraction of time with the highest accuracy which helps to classify the spam and ham.

4.1 Reusability

The written code for this project can be reused as many times possible without any errors or issues.

4.2 Application Compatibility

The main component in the project will be using Python for writing the code and web app which helps to retrieve and submit the data from the database over the internet.

4.3 Resource Utilization

When the input is given, it will take the necessary actions to get accurate results from the provided information.

4.4 Deployment

5 Conclusion

The Designed web application for Spam Ham Classifier will help to classify the spam and ham with the help of the dataset provided.

6 References

1. https://www.securelist.com/en/analysis/204792230/Spam_Report_April_2012
2. http://telco-soft.in/mailserver.php
3. http://www.csmining.org/index.php/malicious-software-datasets-.html
4. http://www.mathworks.com/help/phased/examples/detector-performance-analysis-using-roccurves.html
5. https://www.analyticsvidhya.com/blog/2015/09/naive-bayes-explained/