HIGH LEVEL DESIGN(HLD)

SPAM-HAM CLASSIFIER

Revision Number:1.0
Last date of revision:14/06/2022

Table of Contents

1. Document Version Control

Date Issued	Version	Description	Author
14/06/2022	1.0	Initial HLD	Animesh Jain

2. Abstract

The increased number of unsolicited emails known as spam has necessitated the development of increasingly reliable and robust antispam filters. Recent machine learning approaches have been successful in detecting and filtering spam emails. I need 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 being ham (legitimate) or spam.

3. Introduction

3.1 Why this High-Level Design?

The purpose of this High-Level Design (HLD) Document is to add the important details about this project. Through this HLD Document, I'm going to describe every small and big-things about this project.

4. General Description

4.1 Product Perspective

The Spam-Ham Classifier predicts the message using classification based machine learning algorithm that is Multinomial Naive Bayes.

4.2 Problem Statement

The increased number of unsolicited emails known as spam has necessitated the development of increasingly reliable and robust antispam filters. Recent machine learning approaches have been successful in detecting and filtering spam emails. I need 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 being ham (legitimate) or spam.

4.3 Proposed Solution

The solution is to build a machine learning algorithm which will be able to classify the message. We have many classification based algorithm like Logistic Regression, Decisiontree classifier, Random forest Classifier, XGBClassifier etc.

Upon experimentation on the data I got to know that Multinomial Naïve Bayes is performing way better than all other classifier algorithms. We are going to pick up only MNB because of its good accuracy. But before that we are going to preprocess the raw data provided by our client and then the model building process will come.

4.4 Technical Requirements

In this project we are having a set of requirements and they are given below

a)Model should be exposed through API or User Interface, so that anyone can test model.

b) Model should be deployed on cloud (Azure, AWS, GCP).

c) Cassandra database should be integrated in this project for any kind of user input.

4.5 Data Requirements

Data Requirement completely depend on our problem.

a) For training and testing the model, we are using spam-ham classifier dataset from UCI machine learning repository.

b) Dataset Link: https://archive.ics.uci.edu/ml/datasets/sms+spam+collection

c) From user we are taking just the message in a text format:

4.6 Tools Used

scikit
learn

4.7 Constraints

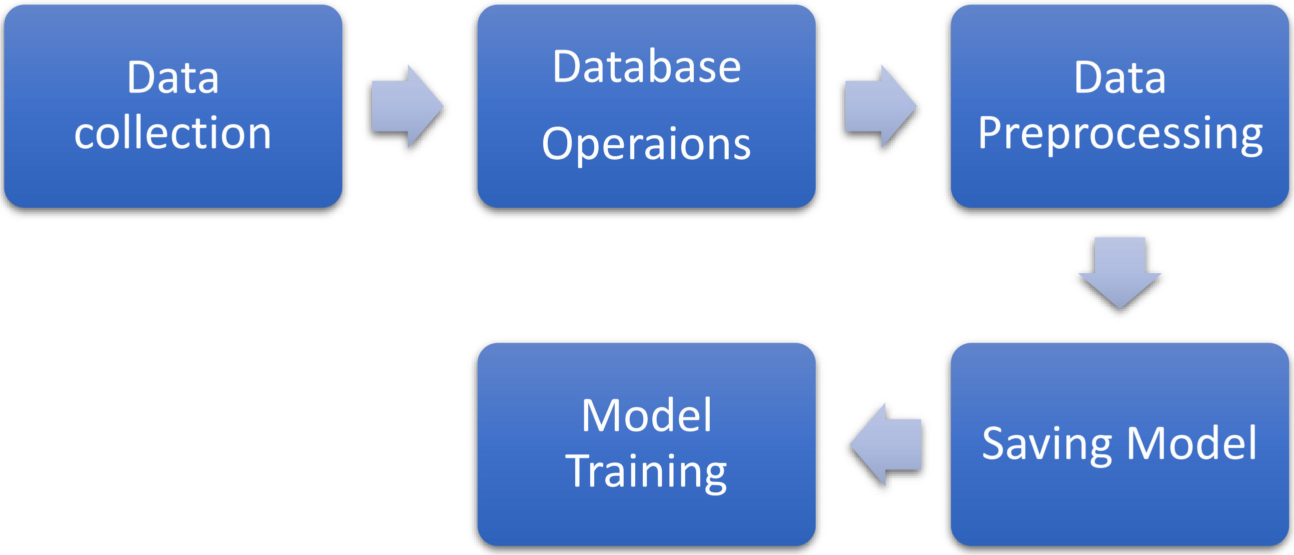
The Spam-Ham Classifier system must be user friendly, errors free and users should not be required to know any of the back-end working.

4.8 Assumptions

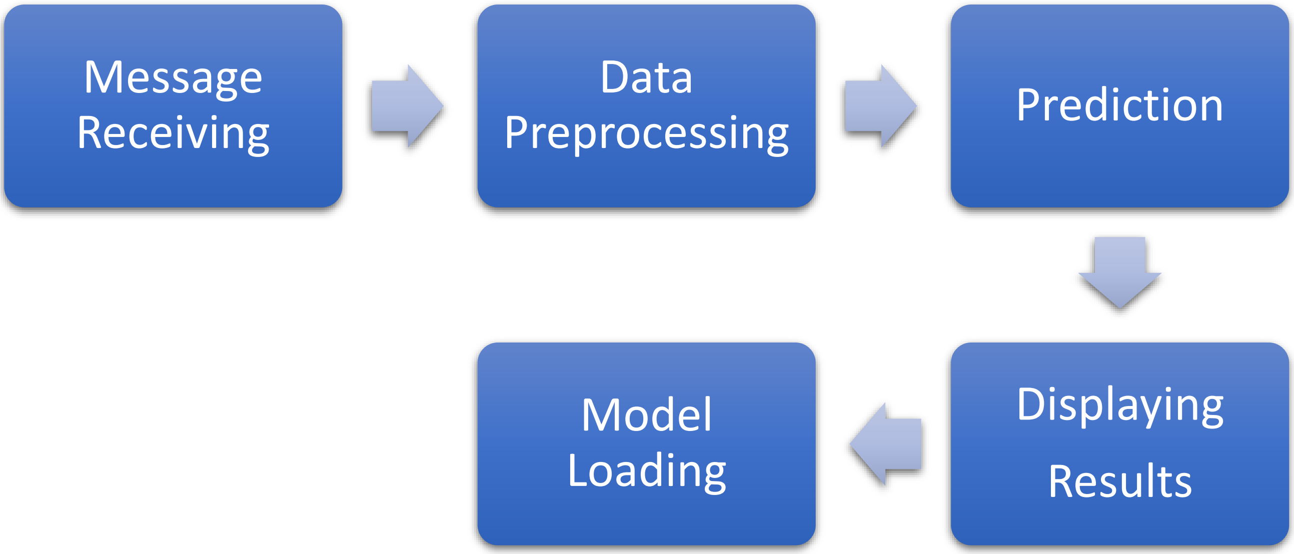
It is assumed that all the aspect of this project have the ability to work together in the way designer is expecting.

5. Design Details

5.1 For Training



5.2 Deployment process



5.3 Event Log

In this Project we are logging every process so that the user will know what process is running internally. We have designed logging in such a way that debugging will be an easy task .

5.4 Error Handling

We have designed this project in such a way that, at any step if error occur then our application should not terminate rather it will catch that error and display that error with proper explanation as to what went wrong during process flow.

6. Performance

Solution of Spam-Ham Classifier is used to classify the message type in advance, so it should be as accurate as possible so that it should give as much as possible accurate classification.

6.1 Reusability

We have done programming of this project in such a way that it should be reusable. So that anyone can add and contribute without facing any problems.

6.2 Application Compatibility

The different module of this project is using Python as an interface between them. Each module will have it's own job to perform and it is the job of the Python to ensure the proper transfer of information.

6.3 Resource Utilization

In this project, when any task is performed, it will likely that the task will use all the processing power available in that particular system until it's job finished.

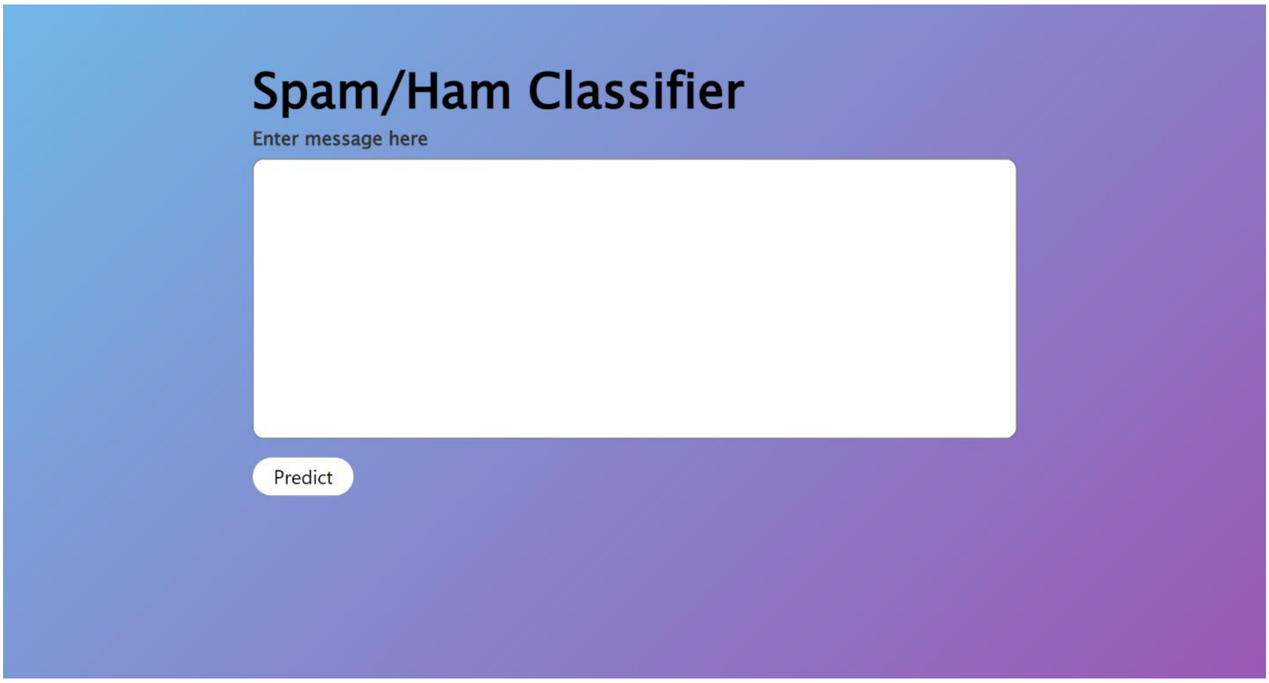
6.4 Deployment

I am deploying my model into GCP and the link is

http://ec2-3-140-241-66.us-east-2.compute.amazonaws.com:5000/

6.5 User Interface

We have created an UI for user by using HTML and CSS.



7. Conclusion

The Spam-Ham Classifier model will classify the message type in prior so that people can be safe from frauds, unwanted messages and it will also save the time of people by classifying the message which are important.