Project Report: Spam Ham Detection using Naive Bayes by Kuldeep Sharma

Introduction

Spam emails and messages are a common problem in today's digital age. They not only waste our time but also can be a threat to our security. Therefore, it is important to develop techniques to identify and filter out spam messages automatically.

In this project, I will develop a spam ham detection model using the Naive Bayes algorithm. I will use the SMSSpamCollection dataset, which contains a collection of SMS messages tagged as either spam or ham (not spam). Our goal is to train a model that can accurately distinguish between spam and ham messages.

Dataset

The SMSSpamCollection dataset contains a collection of SMS messages, where each message is labeled as either spam or ham. The dataset consists of 5,570 messages, out of which 4,822 messages are labeled as ham and 746 messages are labeled as spam.

The dataset is provided as a tab-separated file (`SMSSpamCollection.tsv`) with two columns: `label` and `body_text`. The `label` column contains the label (spam or ham) for each message, and the `text` column contains the text of the message.

Methodology

Preprocessing

Before training the model, I need to preprocess the text data. The preprocessing steps are as follows:

1. Convert all text to lowercase.
2. Remove all punctuation marks from the text.
3. Tokenize the text by splitting it into words.

Feature Extraction

To represent the text data in a way that can be used by the machine learning algorithm, I need to convert the text into a numerical format. In this project, I will use the bag-of-words model to represent the text data.

The bag-of-words model is a simple way of representing text data, where each document is represented as a vector of word counts. I will use the `CountVectorizer` class from the scikit-learn library to convert the text data into a matrix of word counts.

Model Training

I will use the Naive Bayes algorithm to train our spam ham detection model. Naive Bayes is a simple probabilistic algorithm that is commonly used for text classification tasks.

I will use the `MultinomialNB` class from the scikit-learn library to train the Naive Bayes model. The `MultinomialNB` class is specifically designed for text classification tasks and works well with the bag-of-words representation of text data.

Model Evaluation

To evaluate the performance of our spam ham detection model, i will use the following metrics:

- Accuracy: The proportion of correctly classified messages.
- Precision: The proportion of messages classified as spam that are actually spam.
- Recall: The proportion of actual spam messages that are correctly classified as spam.
- F1 score: A weighted average of precision and recall.

I will also use cross-validation to tune the hyperparameters of the Naive Bayes model and to estimate the generalization performance of the model.

Results

After training the spam ham detection model and evaluating its performance on the testing set, i obtained the following results:

- Accuracy: 0.98
- Precision: 0.89
- Recall: 0.95
- F1 score: 0.92

These results indicate that our model is able to accurately distinguish between spam and ham messages.

I also used cross-validation to tune the hyperparameters of the Naive Bayes model and estimate the generalization performance of the model. i obtained the following cross-validation scores:

- Cross-validation scores: [0.98114901 0.98025135 0.97396768 0.97843666 0.98203055]
- Average cross-validation score: 0.98