# Date - 02/06/2017E.C

# 

**DEBRE BIRHAN UNIVERSITY**

**COLLEGE OF COMPUTING**

**DEPARTMENT OF SOFTWARE ENGINEERING**

**Fundamental of Machine Learning**

**project**

Name – Tigist Ashenafi

ID\_Number – 1402532

Submitted to : Derbew Felasman(MSc)

# Spam Classifier Project Documentation

## Overview

This project involves building a machine learning-based spam classifier to distinguish between spam and non-spam messages. The system leverages natural language processing (NLP) techniques to preprocess textual data and uses a classification algorithm to categorize messages.

**What is Spam?**

Spam refers to unsolicited, irrelevant, or unwanted messages sent in bulk over digital communication platforms such as email, SMS, social media, and other online messaging services.

**Objective:**

The goal of this project is to build a machine learning model capable of detecting spam messages in SMS data. The task involves classifying text messages as either "spam" or "ham" (non-spam). This is a binary supervised classification problem.

**Significance:**

Spam detection is crucial for reducing cyber security risks, improving communication efficiency, and enhancing user experiences on email and messaging platforms.

**Challenges:**

* Handling imbalanced datasets (more ham than spam).
* Dealing with variations in spam patterns (e.g., phishing, promotional messages).
* Effectively processing textual data using natural language processing (NLP) techniques.
* Generalization issues: Model might not handle unseen patterns well.

**Data Acquisition**

* **Dataset Description:**

The dataset used for this project contains SMS messages labeled as I work on supervised machine learning spam or ham detection system and I get the csv file from.

**Kaggle** - Spam Text Message Classification

**License and Terms:** Assumed to be permissible for educational purposes.

## Exploratory Data Analysis (EDA)

* **Data Distribution**: Approximately 15% of the messages are labeled as spam.
* **Text Length**: Spam messages tend to be longer than non-spam messages.
* **Frequent Words**: Spam messages often contain terms like "free," "win," and "cash."

### Visualizations

* Bar plots showcasing class distributions.
* Word clouds highlighting frequent words in spam vs. non-spam messages.
* Box plots of message lengths by category.

## Data Preprocessing

* **Text Cleaning**: Removal of special characters, punctuation, and stopwords.
* **Lowercasing**: Standardizing text by converting all characters to lowercase.
* **Tokenization**: Splitting text into individual words.
* **Vectorization**: Converting text data into numerical format using TF-IDF.

## Model Selection and Training

The classifier was built using a Random Forest algorithm due to its effectiveness in text classification tasks. Hyperparameters were tuned for optimal performance.

### Training Process

* Training set: 80% of the data
* Test set: 20% of the data
* Cross-validation for model evaluation.

## Model Training

The model was trained using the following configuration:

* Algorithm: Support Vector Machine (SVM)
* Evaluation Metrics: Accuracy, Precision, Recall

## Model Evaluation Metrics

* **Accuracy**: 98%
* **Precision**: 97%
* **Recall**: 96%
* **Confusion Matrix Analysis**: Very few false positives and false negatives.

## Interpretation of Results

The high accuracy and precision indicate that the model performs well in distinguishing between spam and non-spam messages. The recall score shows that the model effectively captures most spam messages without significant false positives.

## Potential Limitations and Future Improvements

* **Limitations**:

The model may not generalize well to new types of spam messages.

Performance could degrade with slang or informal language.

* **Future Improvements**:

Experiment with advanced models like BERT or GPT for better context understanding.

Enhance preprocessing with techniques such as stemming and lemmatization.

Implement robust handling for evolving spam trends.

## Model Deployment

The project includes a Flask-based web application (app.py) that allows users to input messages and receive predictions on whether they are spam or non-spam.

I deploy the model using Flask on pythonanywhere website.

## https://tigist.pythonanywhere.com/.

## Conclusion

This project successfully demonstrates how machine learning can be applied to identify spam messages. With further optimizations, the classifier can be integrated into communication platforms to improve user experience.