

Spam SMS Detection

1. Introduction:

The project focuses on developing a Spam Detection Model using Logistic Regression to classify SMS messages as either “Spam” or “Ham” (legitimate). The model is trained on a labelled dataset of SMS messages and aims to accurately detect unsolicited or fraudulent messages. It is an end-to-end NLP-based machine learning project that involves text preprocessing, feature extraction using TF-IDF vectorization and classification.

The objective of this project is to develop an SMS Spam Classifier that automatically distinguishes between spam and legitimate messages (referred to as “ham”). Using advanced Natural Language Processing (NLP) and Machine Learning (ML) techniques, this project aims to create a reliable and efficient classification system. The system is trained on a dataset containing thousands of labelled SMS messages, categorized as either “spam” or “ham”. By applying a series of data cleaning, preprocessing, and vectorization techniques such as TF-IDF, we convert unstructured text data into meaningful numerical representations suitable for machine learning. The project is designed with practicality in mind, allowing business or individuals to integrate the model into their SMS processing systems for automatic spam detection, improving security and user experience.

2. Requirements:

→Hardware:

1. A system with at least 4 GB RAM and 32-bit system
2. Python-Compatible environment

→Software/Packages:

1. Python version 3
2. Pandas
3. Scikit-Learn
4. Matplotlib
5. Seaborn
6. Pickle
7. Google Co-lab
8. Microsoft Excel

3. Technologies Used:

- **Programming Language:** Python
- **Machine Learning Algorithm:** Logistic Regression
- **Vectorization Technique:** TF-IDF (Term Frequency-Inverse Document Frequency)
- **Python Libraries:** Pandas, NumPy, Matplotlib, Seaborn, Scikit-Learn, Pickle
- **Development Tools:** Google Co-lab, MS-Excel, Git/GitHub.

4. Working of the Model (Step-by-Step Procedure)

Step-I: Data Loading

- Load the provided dataset (spam.csv) using pandas' library

Step-II: Data Cleaning

- Removed unnecessary columns like metadata.
- Renamed columns to 'label' and 'message'.
- Converted labels into binary format (1=Spam,0=Ham)

Step-III: Text Pre-Processing

- Converted messages to lowercase
- Removed special characters and stopwords
- Vectorized messages using Term Frequency-Inverse Document Frequency (TF-IDF).

Step-IV: Model Training and Testing

- Split the data into training and test sets (75-25) ratio.
- Trained a logistic regression model using scikit-learn library

Step-V: Model Evaluation

- Evaluated model using **accuracy score**, **classification report**, **precision score** and **confusion matrix**.
- Achieved high accuracy with well-balanced precision and recall.

Step-VI: Save the Model

- Saved the trained logistic regression model and TF-IDF vectorizer using pickle for deployment or reuse in the other projects for spam messages detection.

5. How to Use the Model

Step 1: Load the saved model and vectorizer:

```
import pickle
```

```
# Load model
```

```
with open ('spam_classifier_logistic.pkl', 'rb') as model_file:
```

```
    model = pickle.load(model_file)
```

```
# Load vectorizer

with open('tfidf_vectorizer.pkl', 'rb') as vec_file:

    vectorizer = pickle.load(vec_file)
```

Step 2: Make Predictions:

```
sample = ["Your account has been blocked. Verify immediately!"]

sample_vectorized = vectorizer.transform(sample)

prediction = model.predict(sample_vectorized)

print("SPAM" if prediction[0] == 1 else "HAM")
```

6. Features

- Preprocesses and cleans raw SMS text
- Uses TF-IDF for efficient vectorization
- High-accuracy spam detection using Logistic Regression
- Supports real-time prediction on new SMS inputs
- Visualizes data insights (EDA)
- Model and vectorizer are saved for easy reusability
- Lightweight and easy-to-deploy

7. Conclusion:

This project successfully demonstrates how machine learning and NLP techniques can be applied to detect SMS spam. The model achieves a balance between accuracy and computational efficiency using Logistic Regression combined with TF-IDF. This solution is suitable for SMS filtering tasks and can be integrated into real-world systems with minor adjustments.

8. Project Access Link:

GitHub Link: <https://github.com/md-ashraf98/Spam-SMS-Detection>