**SMS Spam Detection System Documentation**

**Introduction**

The SMS Spam Detection System is designed to classify text messages as either "spam" or "not spam" (ham). This system is built using machine learning algorithms trained on a dataset containing labeled examples of SMS messages.

**Components**

The system consists of the following components:

**Data Preprocessing**

* Model Training
* Model Evaluation
* Model Deployment
* Data Preprocessing

**Overview**

Data preprocessing involves cleaning and transforming the raw SMS data into a format suitable for machine learning models.

**Steps**

* Loading Data: The SMS dataset is loaded into a Pandas DataFrame.
* Cleaning Text: Text preprocessing techniques such as converting to lowercase, removing special characters, digits, links, and stopwords, tokenizing, and stemming are applied to clean the text messages.
* Balancing Dataset: Since the dataset is imbalanced, where the majority of messages are labeled as "not spam," the dataset is balanced using the RandomOverSampler from the imbalanced-learn library.

**2. Model Training**

**Overview**

Three different machine learning algorithms are trained on the preprocessed data to classify SMS messages.

**Algorithms Used**

**Random Forest Classifier**

**Logistic Regression Classifier**

Support Vector Machine (SVM) Classifier

Steps

* Train-Test Split: The dataset is split into training and testing sets.
* TF-IDF Vectorization: Text data is converted into numerical feature vectors using TF-IDF vectorization.
* Model Training: Each classifier is trained on the training data.

**3. Model Evaluation**

Overview

The performance of each trained model is evaluated using metrics such as accuracy, precision, recall, and F1-score.

Metrics Used

* Confusion Matrix: Provides a summary of correct and incorrect predictions made by each model.
* Classification Report: Presents precision, recall, F1-score, and support for each class (spam and not spam).

**4. Model Deployment**

Overview

* The trained Random Forest Classifier is deployed as the final model for detecting spam messages.
* Create flask based user interface where we have two option either upload file or type manually text and our system detect it by either fake or real

Steps

* Saving Model and TF-IDF Vectorizer: The trained Random Forest model and TF-IDF vectorizer are serialized using pickle and saved to disk.
* Loading Saved Model: The saved model and TF-IDF vectorizer can be loaded for making predictions on new text messages.

**Usage**

Detecting Spam Messages

The predict\_fake\_or\_real function is provided for detecting spam messages. Simply pass a text message as input, and it will return whether the message is "fake" (spam) or "real" (not spam).