**SMS Spam Classifier using Logistic Regression**  
**Aim:**  
To build a machine learning model using Logistic Regression that can classify SMS messages as "spam" or "ham" (not spam) based on their content.

**Abstract:**  
With the growing volume of mobile communications, spam detection has become essential for user safety and convenience. This project applies a basic supervised machine learning algorithm—**Logistic Regression**—to classify SMS messages as spam or not spam using the **SMS Spam Collection Dataset**. The model is trained on text data by extracting features through natural language processing techniques such as TF-IDF vectorization. This project provides a beginner-friendly introduction to classification problems and showcases the practical application of logistic regression in real-world scenarios like spam detection.

**Working Flow of the Algorithm:**

**1. Data Collection:**  
The dataset (SMS Spam Collection) is loaded from a .csv or .tsv file containing SMS messages labeled as **'spam'** or **'ham'**.

**2. Data Preprocessing:**

* Text messages are cleaned (e.g., lowercased, punctuation removed).
* Labels are encoded into binary values.
* Messages are transformed into numerical features using **TF-IDF vectorization**.

**3. Model Building:**  
Use **LogisticRegression** from sklearn.linear\_model to train the model on the extracted features.

**4. Prediction:**  
The trained model predicts whether a given SMS message is spam or not.

**5. Evaluation:**  
Evaluate model performance using accuracy, precision, recall, F1-score, and confusion matrix.

**Sample Code Outline:**

python

CopyEdit

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report, confusion\_matrix

# Load dataset

df = pd.read\_csv("spam.csv", encoding='latin-1')[['v1', 'v2']]

df.columns = ['label', 'message']

# Encode labels

df['label'] = df['label'].map({'ham': 0, 'spam': 1})

# Feature extraction

tfidf = TfidfVectorizer(stop\_words='english')

X = tfidf.fit\_transform(df['message'])

y = df['label']

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Predict and evaluate

predictions = model.predict(X\_test)

print(confusion\_matrix(y\_test, predictions))

print(classification\_report(y\_test, predictions))

# Predict example

sample = tfidf.transform(["Congratulations! You've won a free ticket. Call now!"])

print("Prediction:", "Spam" if model.predict(sample)[0] else "Ham")

**Output Example:**

**Input:** "Congratulations! You've won a free ticket. Call now!"  
**Prediction:** Spam

**Metrics:**

* Accuracy: 0.98
* Precision: 0.97
* Recall: 0.96
* F1-score: 0.96

**Confusion Matrix:**

lua

CopyEdit

[[966 5]

[ 13 131]]



**Result:**  
The model accurately classifies spam messages using simple logistic regression and TF-IDF features. It achieves high accuracy and performs well on both precision and recall metrics, proving that even basic ML techniques can effectively solve text classification problems.