SMS Spam Detection Project - Instructions

# Project Overview

This project automatically classifies SMS messages as spam or ham using Natural Language Processing (NLP) and Machine Learning. It includes data preprocessing, feature extraction, model training, evaluation, and visualization of results.

# Folder Contents

- spam.csv: Dataset containing SMS messages and labels (spam/ham).
- SMS\_Spam\_Detection.ipynb: Jupyter Notebook containing all scripts.
- README.txt / INSTRUCTIONS.txt: Instruction file.
- Documentation.pdf (optional): Detailed project workflow, EDA, visualizations, and results.

# Required Libraries

Before running the project, make sure the following Python libraries are installed:

- pandas: Data manipulation
- numpy: Numerical computations
- matplotlib: Data visualization
- seaborn: Statistical data visualization
- nltk: Natural Language Processing (tokenization, stopwords, stemming)
- scikit-learn: Machine learning algorithms and feature extraction
- wordcloud: Word cloud visualization

Installation command:
 pip install pandas numpy matplotlib seaborn nltk scikit-learn wordcloud

# Instructions to Run the Project

1. Open the Project: Open the Jupyter Notebook SMS\_Spam\_Detection.ipynb in your IDE. Ensure spam.csv is in the same folder.

2. Data Preprocessing: The notebook will remove duplicates, handle missing values, and encode target labels.

3. Text Preprocessing: Messages will be converted to lowercase, tokenized, cleaned by removing stopwords/punctuation, and stemmed.

4. Feature Extraction: Text is converted into numerical features using TF-IDF Vectorization. Features are scaled and split into training and testing sets.

5. Model Training & Evaluation: Train Naive Bayes classifiers (Gaussian, Multinomial, Bernoulli) and evaluate performance using accuracy, precision, and confusion matrix.

6. Visualizations (Optional): WordClouds for spam and ham messages. Histograms, pie charts, and correlation heatmaps.

7. Results Analysis: Review metrics to understand model performance and identify which words/patterns indicate spam.

# Notes & Recommendations

- You can try other ML models like Logistic Regression or Random Forest for better accuracy.
- Extend this project into a real-time SMS spam detection system.
- Useful for learning NLP techniques, text preprocessing, and machine learning model evaluation.