UIET CSJM UNIVERSITY KANPUR

Capstone Project

(CAP – S101)

SMS SPAM CLASSIFIER

SYNOPSIS FILE

CSE – AI (2K22)

SUBMITTED BY -
PRANSHU YADAV
CSJMA22001390327

SUBMITTED TO -
DR. SANJAY SINGH

1. Title of Project:

- The goal is to build a model that can classify SMS messages as either "spam" or "ham" (non-spam). Spam messages often contain certain patterns, keywords, or styles that differ from regular messages.

2. Dataset:

- You can use publicly available datasets, such as:
  - SMS Spam Collection Dataset from the UCI Machine Learning Repository.
  - Kaggle also has many spam SMS datasets.
- The dataset typically consists of labeled messages (`spam-->1` or `ham-->0`), which makes it suitable for supervised learning.

3. Data Preprocessing:

- **Text cleaning:**
  - Remove special characters, numbers, and punctuation.
  - Convert all text to lowercase.
  - Tokenize the text into words.
  - Remove stop words (common words like "the", "and", etc.).

Feature extraction:

- Convert the text data into numerical form (vectors) using techniques like:
  - **TF-IDF (Term Frequency - Inverse Document Frequency):** Weighs words based on their importance in the document compared to the whole corpus.
  - **Bag of Words:** Represents text by counting the occurrence of words in the corpus.

4. Model Selection:

You can experiment with several machine learning models:

- **Naive Bayes:** Often works well for text classification tasks like spam detection.
- **Logistic Regression:** A simple yet effective classification model.
- **Support Vector Machine (SVM):** Great for text classification tasks, especially in high-dimensional spaces.
- **Random Forest or Decision Trees:** These models can handle complex decision boundaries.

5. Handling Imbalanced Data:

- In many real-world spam detection scenarios, spam messages are less frequent than regular messages. You can handle this imbalance using techniques like:
    - **Resampling:** Over-sampling the minority class or under-sampling the majority class.
    - **Synthetic Data Generation:** Using algorithms like SMOTE to generate synthetic examples of the minority class.

6. Model Deployment (Optional for Capstone):

- Once you have a trained model, you can build a web or mobile application to deploy it. For example:
    - **Streamlit:** For creating an interactive dashboard.

7.Documentation:

- Clearly document your process, from data collection and cleaning to model selection and evaluation.

- Ensure that your project is well-commented, and include visualizations to explain the model's performance.

Example Structure of the Capstone:

1. **Introduction:**
    a. Brief explanation of the problem, importance of spam detection, and project goals.
2. **Dataset Exploration:**
    a. Provide details about the dataset, its structure, and how the data is preprocessed.
3. **Model Building:**
    a. Detailed explanation of the models you tried, including any hyperparameter tuning.
4. **Evaluation:**
    a. Present the results of your model evaluation, using confusion matrices and performance metrics.
5. **Conclusion:**
    a. Summarize your findings and any limitations. Optionally, suggest future work or improvements.