1. **Downloaded data from Kaggle (SMS Spam Collection Dataset).**
2. **Data Cleaning**

* Removed unwanted columns (Unnamed: 2, Unnamed: 3, Unnamed: 4).
* Renamed columns: v1 to Targets and v2 to Text.
* Converted the Targets column to binary values: ham = 0 and spam = 1.
* Removed duplicate values.

1. **Exploratory Data Analysis (EDA)**

* Analyzed the data distribution.
* Created three new columns:  
  + num\_characters: Number of characters in each SMS/email.
  + num\_words: Number of words in each SMS/email.
  + num\_sentences: Number of sentences in each SMS/email.
* Plotted bar graphs for visual representation.

1. **Data Preprocessing**

* Created a function named transformed\_text to prepare data for prediction:  
  + Converted text to lowercase.
  + Applied tokenization (breaking text into tokens).
  + Removed special characters (e.g., @, $, &).
  + Eliminated stopwords (e.g., "I," "you") and punctuation (e.g., ".", ",").
  + Applied stemming (e.g., "loving" → "love").
* Used WordCloud to visualize frequent words in spam and ham messages.
* Identified the top 30 ham and spam words for better understanding.

1. **Model Building**

* Vectorized the text column (message content).
* Applied train\_test\_split on the data (divided the data into 80% for training and 20% for testing).

* Evaluated models:  
  + Naive Bayes variants (GaussianNB, MultinomialNB, BernoulliNB) – calculated accuracy and precision scores.
  + Compared with other classifiers:  
    - Logistic Regression, SVC, DecisionTreeClassifier, KNeighborsClassifier, RandomForestClassifier, AdaBoostClassifier, BaggingClassifier, ExtraTreesClassifier, GradientBoostingClassifier, XGBClassifier.
  + Best-performing models: KNeighborsClassifier, Naïve Bayes, and ExtraTreesClassifier.
* Further refined the approach:  
  + Limited features to the top 3,000 most frequent words.
  + Selected Multinomial Naive Bayes as the best-performing model.
* Also experimented with:  
  + Voting Classifier (KNeighborsClassifier, Naïve Bayes, ExtraTreesClassifier) – results were not as good as Naive Bayes.
  + Stacking (with RandomForestClassifier as meta-learner) – did not outperform Naive Bayes.

1. Deployment

* Built a web app using the Streamlit library.
* Deployed the application on Streamlit Cloud.