Project Name = Email classification model

1. Introduction

The goal of this project is to develop a machine learning pipeline to classify emails as spam or ham (not spam) using natural language processing (NLP). The solution involves data preprocessing, exploratory data analysis (EDA), text cleaning using NLTK, and visualization to understand the dataset distribution. Used different ML model (Logistic regression, KNN, XG, ANN)

- The Spam Email Detection project begins by reading the dataset (spam.csv) using the Pandas library. After loading, we inspect the dataset using df.info() to understand the structure, data types, and any missing values, and we preview a few records with df.head(). This allows us to confirm that the dataset includes necessary columns like text and label. A null value check is performed using df.isnull().sum() to ensure data integrity. Once confirmed, we perform exploratory data analysis (EDA) to understand the distribution of spam and ham messages.

- The Spam Email Detection project begins by reading the dataset (spam.csv) using the Pandas library. After loading, we inspect the dataset using df.info() to examine the structure, data types, and non-null entries, and use df.head() to preview the data. A null value check is then performed using df.isnull().sum() to ensure there are no missing values that could affect analysis. Following the initial inspection, we proceed with data visualization to better understand the dataset. A pie chart is plotted to display the distribution of spam and ham messages, helping us detect any class imbalance. Then, we calculate the length of each email message and plot a histogram to visualize how text length varies between spam and ham emails, which may indicate patterns related to spam detection. Additionally, we compute a correlation heatmap for any numeric features (such as email length) to explore potential relationships between variables. These visualizations provide intuitive insights into the dataset before model training.

- After the EDA, we apply text preprocessing using NLTK, which involves converting all text to lowercase, tokenizing the text into individual words, removing punctuation, and filtering out common English stopwords. The cleaned version of each email is saved in a new column called clean_text. This preprocessing prepares the data for vectorization and model training, ensuring it is clean, normalized, and machine-readable.

- After preprocessing the text data, we proceed to the modeling phase, which includes both traditional machine learning algorithms and a deep learning model. For machine learning, we build and evaluate Logistic Regression, XGBoost, and K-Nearest Neighbors (KNN) classifiers. Each model is trained on the cleaned and vectorized text data (using methods such as TF-IDF), and predictions are made on the test set. In parallel, we also construct a Deep Neural Network (ANN model) using Keras, with multiple dense layers, ReLU activation, and a final sigmoid output layer for binary classification. All models are evaluated using standard metrics, including the confusion matrix, recall, F1-score, and the classification report (which includes precision, recall, F1-score, and support). The confusion matrix provides a visual breakdown of true positives, false positives, true negatives, and false negatives. Among all models tested, Logistic Regression and XGBoost achieved the best performance, yielding high accuracy and balanced F1-scores, indicating they are well-suited for spam classification in this dataset. The results validate the effectiveness of both linear and tree-based approaches for text classification, with deep learning models being competitive but more computationally intensive.