Report

Introduction:

Internship Code for Building a Spam Detection Model using TensorFlow and Keras .This internship code confidently creates a binary text classification model using TensorFlow and Keras to differentiate between spam and non-spam (ham) messages. The primary objective of this project is to establish a solid foundation for a more robust spam detection system.

Import the libraries:

For spam\_ham dataset we predicted by using some libraries are TensorFlow, Pandas, NumPy and scikit-learn. These libraries are used for various functions, including data manipulation, model building, and evaluation.

load dataset:

The dataset is loaded from a CSV file named "spam\_ham\_dataset.csv" using Pandas . Name it as spam. Spam.head() gives first five records . The spam.tail() gives last five records. Information and describe the data.This database contains 4 attributes are Unnamed,label,text,label\_num.

Remove Unnecessary columns :

('Unnamed: 0' and 'label') are dropped from the DataFrame.

Splitting the data:

The input features (x) and labels (y) are separated from the Data Frame, and the dataset is split into training and testing sets using train\_test\_split().We partitioned the dataset into a 80:20 ratio, allocating 80% for training and 20% for testing. This split ensures sufficient data for learning while retaining a substantial subset for an unbiased evaluation of the model's performance.

Tokenization:

Tokenization and Padding are then applied to the training text data (X\_train) using Tokenizer from Keras. Padding ensures a uniform length of sequences using pad sequences. Next, a Sequential model is created using Keras, consisting of an Embedding layer for word embeddings, a Flatten layer to flatten the input, followed by Dense layers with ReLU and sigmoid activations for classification. The model is compiled using the Adam optimizer and binary cross-entropy loss function. The model is then trained using model.fit() on the padded training data (X\_train\_pad) and corresponding labels (Y\_train) for 5 epochs with a validation split of 0.2. Predictions are made on the test data (x\_test\_pad) using model.predict(). Predictions are thresholded at 0.5 to obtain binary predictions.

Conlusion:

The binary predictions are displayed. This code confidently implements a basic text classification model using TensorFlow and Keras. With further refinement and evaluation, this code has the potential to serve as a strong foundation for building more robust spam detection systems. This code confidently implements a basic text classification model using TensorFlow and Keras. With further refinement and evaluation, this code has the potential to serve as a strong foundation for building more robust spam detection systems.