**Project Title: Spam Detection Using TensorFlow**

**Introduction:**

The objective of this project was to develop a spam detection system using TensorFlow, a popular open-source machine learning framework. The project aimed to distinguish between spam and legitimate emails with a high level of accuracy. The following report outlines the steps taken, the results obtained, and the conclusions drawn from the analysis.

**Steps Taken:**

**Data Collection:**

Acquired a diverse dataset containing labelled spam and ham (legitimate) emails.

Ensured the dataset represented real-world scenarios to enhance the model's practicality.

data**=**pd**.**read\_csv("spam\_ham\_dataset.csv")

Dataset Used: [Spam Mails Dataset (kaggle.com)](https://www.kaggle.com/datasets/venky73/spam-mails-dataset)

**Data Preprocessing:**

Conducted extensive data cleaning, removing any irrelevant or duplicate entries.

Utilized tokenization to break down textual data into meaningful units.

Applied sequence padding to ensure uniform input lengths for the model.

data**.**drop("Unnamed: 0", axis**=**1, inplace**=True**)

data**.**drop("label\_num", axis**=**1, inplace**=True**)

**Model Architecture:**

Designed a deep neural network using TensorFlow, incorporating Embedding, Dense, and Dropout layers.

Experimented with various architectures to find the optimal configuration for spam detection.

model **=** Sequential()

model**.**add(Embedding(vocab\_size, 24, input\_length **=** max\_length))

model**.**add(Flatten())

model**.**add(Dense(500, activation**=**'relu'))

model**.**add(Dense(200, activation**=**'relu'))

model**.**add(Dropout(0.5))

model**.**add(Dense(100, activation**=**'relu'))

model**.**add(Dense(1, activation**=**'sigmoid'))

**Model Training:**

Split the dataset into training and testing sets to assess the model's generalization.

Employed optimization techniques, including early stopping, to prevent overfitting.

Trained the model on the training set, iterating through multiple epochs.

x**=**data['text']**.**values

y**=**data['label']**.**values

x\_train, x\_test, y\_train, y\_test**=** train\_test\_split(x, y, test\_size**=**0.2, random\_state**=**42)

In [32]:

**Evaluation:**

Tested the model on the separate testing set to assess its accuracy and performance.

Recorded key metrics, such as precision, recall, and F1 score, to comprehensively evaluate model effectiveness.

Classification Report

precision recall f1-score support

0 0.97 0.98 0.98 742

1 0.95 0.92 0.94 293

accuracy 0.97 1035

macro avg 0.96 0.95 0.96 1035

weighted avg 0.97 0.97 0.97 1035

Accuracy : 0.9652173913043478

**Analysis of Results:**

Achieved an impressive accuracy of 96.52% on the testing set.

Analysed precision and recall to understand the model's ability to correctly identify spam and ham emails.

**Conclusions:**

The model achieved a remarkable accuracy of 96.52%, demonstrating its effectiveness in distinguishing between spam and legitimate emails.

Precision and recall analyses revealed the model's ability to accurately identify spam while minimizing false positives.

The implementation of TensorFlow, combined with careful data preprocessing and model tuning, resulted in a robust and efficient spam detection system.

Ongoing monitoring and potential fine-tuning of the model will ensure its continued effectiveness in real-world scenarios.

The project successfully accomplished its goals, providing valuable insights into the capabilities of TensorFlow for spam detection applications.