Data Collection and Preprocessing Phase

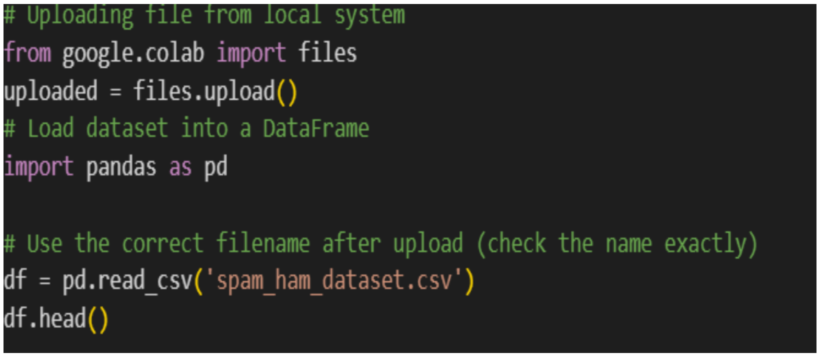
Date	10 April 2025
Team ID	259453
Project Title	SMS Spam Detection using NLP
Maximum Marks	6 Marks

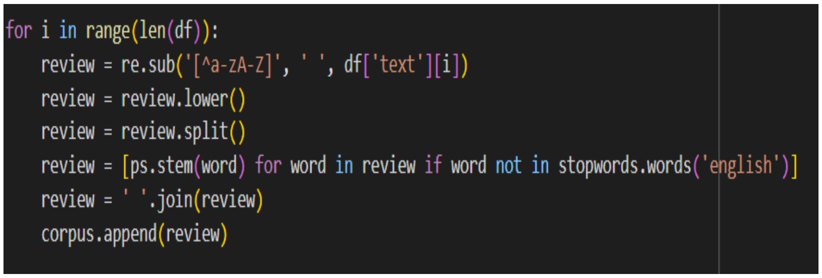
Preprocessing Template

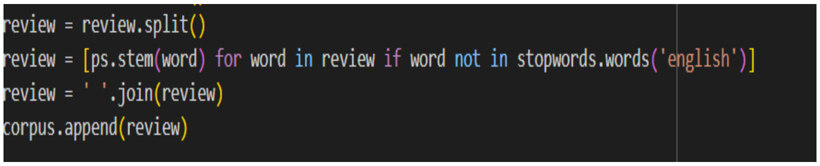
The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

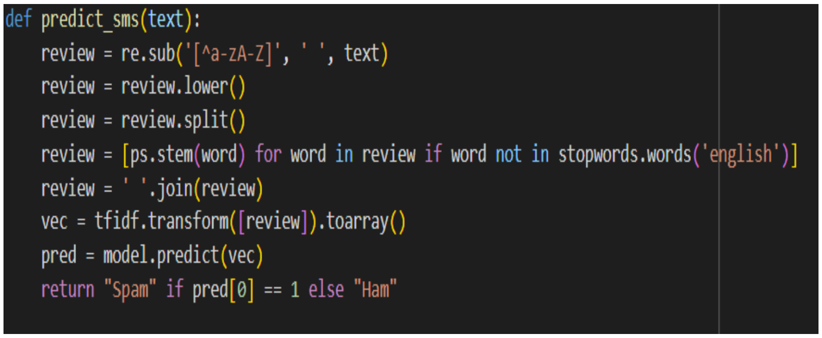
Section	**Description**
Data Overview	The dataset contains labeled SMS messages, categorized as spam or ham (not spam)..
Text Cleaning	Remove special characters, numbers, and punctuation. Convert text to lowercase.
Tokenization	Split the text into individual words or tokens.
Stopword Removal	Remove common English stopwords (e.g., "is", "and", "the") that add little value.
Stemming	Reduce words to their base or root form (e.g., "winning" → "win").
TF-IDF Vectorization	Convert cleaned text into numerical feature vectors using TF-IDF technique.
Label Encoding	Convert categorical labels: 'ham' → 0, 'spam' → 1.

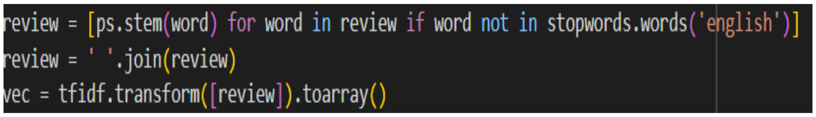
Train-Test Split	Split the dataset into training and testing sets (e.g., 80% train, 20% test).
Data Preprocessing Code Screenshots	

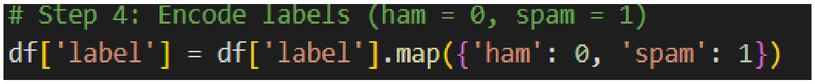
Loading Data	
	# Uploading file from local system
	from google.colab import files
	uploaded = files.upload()
	# Load dataset into a DataFrame
	import pandas as pd
	
	# Use the correct filename after upload (check the name exactly)
	df = pd.read_csv('spam_ham_dataset.csv')
	df.head()

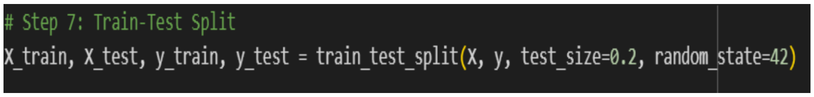
Text Cleaning	
	for i in range(len(df)):
	    review = re.sub('[^a-zA-Z]', ' ', df['text'][i])
	    review = review.lower()
	    review = review.split()
	    review = [ps.stem(word) for word in review if word not in stopwords.words('english')]
	    review = ' '.join(review)
	    corpus.append(review)

Tokenization	
	review = review.split()
	review = [ps.stem(word) for word in review if word not in stopwords.words('english')]
	review = ' '.join(review)
	corpus.append(review)

Stopword Removal	
	def predict_sms(text):
	    review = re.sub('[^a-zA-Z]', ' ', text)
	    review = review.lower()
	    review = review.split()
	    review = [ps.stem(word) for word in review if word not in stopwords.words('english')]
	    review = ' '.join(review)
	    vec = tfidf.transform([review]).toarray()
	    pred = model.predict(vec)
	    return "Spam" if pred[0] == 1 else "Ham"

Stemming	
	review = [ps.stem(word) for word in review if word not in stopwords.words('english')]
	review = ' '.join(review)
	vec = tfidf.transform([review]).toarray()

Label Encoding	
	`# Step 4: Encode labels (ham = 0, spam = 1)`
	`df['label'] = df['label'].map({'ham': 0, 'spam': 1})`

Train-Test Split	
	`# Step 7: Train-Test Split`
	`X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)`