{

"cells": [

{

"cell\_type": "code",

"execution\_count": 5,

"id": "b57f655c-f48b-4008-b0cb-c29099483e53",

"metadata": {},

"outputs": [],

"source": [

"import pandas as pd\n",

"import re\n",

"from nltk.corpus import stopwords\n",

"from nltk.tokenize import word\_tokenize\n"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "b09c497c-a4fe-43ef-a8ec-5b6ac2ffbcba",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" v1 v2 Unnamed: 2 \\\n",

"0 ham Go until jurong point, crazy.. Available only ... NaN \n",

"1 ham Ok lar... Joking wif u oni... NaN \n",

"2 spam Free entry in 2 a wkly comp to win FA Cup fina... NaN \n",

"3 ham U dun say so early hor... U c already then say... NaN \n",

"4 ham Nah I don't think he goes to usf, he lives aro... NaN \n",

"\n",

" Unnamed: 3 Unnamed: 4 \n",

"0 NaN NaN \n",

"1 NaN NaN \n",

"2 NaN NaN \n",

"3 NaN NaN \n",

"4 NaN NaN \n"

]

}

],

"source": [

"#loading the dataset that contains the Spam And ham data\n",

"data = pd.read\_csv('Spam.csv', encoding='ISO-8859-1')\n",

"print(data.head())"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "4e05696f-b0f0-47e3-9e10-95edcad93081",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" label text\n",

"0 ham Go until jurong point, crazy.. Available only ...\n",

"1 ham Ok lar... Joking wif u oni...\n",

"2 spam Free entry in 2 a wkly comp to win FA Cup fina...\n",

"3 ham U dun say so early hor... U c already then say...\n",

"4 ham Nah I don't think he goes to usf, he lives aro...\n"

]

}

],

"source": [

"data = pd.read\_csv('Spam.csv', encoding='ISO-8859-1')\n",

"# Drop unnecessary columns from the dataset\n",

"data.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)\n",

"\n",

"# Rename columns from the dataset\n",

"data.rename(columns={'v1': 'label', 'v2': 'text'}, inplace=True)\n",

"print(data.head())"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"id": "06b05d4b-f24c-4670-a9b9-67d647503f77",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

" label processed\_text\n",

"0 ham go jurong point crazy available bugis n great ...\n",

"1 ham ok lar joking wif u oni\n",

"2 spam free entry wkly comp win fa cup final tkts st ...\n",

"3 ham u dun say early hor u c already say\n",

"4 ham nah dont think goes usf lives around though\n"

]

}

],

"source": [

"\n",

"data = pd.read\_csv('Spam.csv', encoding='ISO-8859-1')\n",

"\n",

"data.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)\n",

"\n",

"data.rename(columns={'v1': 'label', 'v2': 'text'}, inplace=True)\n",

"\n",

"\n",

"# Preprocess text data\n",

"def preprocess\_text(text):\n",

" # Convert to lowercase\n",

" text = text.lower()\n",

" \n",

" # Remove special characters, numbers, and extra spaces\n",

" text = re.sub(r'[^a-zA-Z\\s]', '', text)\n",

" \n",

" # Tokenize the text\n",

" words = word\_tokenize(text)\n",

" \n",

" # Remove stopwords\n",

" stop\_words = set(stopwords.words('english'))\n",

" words = [word for word in words if word not in stop\_words]\n",

" \n",

" # Join the words back into a string\n",

" preprocessed\_text = ' '.join(words)\n",

" \n",

" return preprocessed\_text\n",

"\n",

"# Apply preprocessing to the 'text' column\n",

"data['processed\_text'] = data['text'].apply(preprocess\_text)\n",

"\n",

"# Display the preprocessed data with all columns\n",

"print(data[['label', 'processed\_text']].head()) \n"

]

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3 (ipykernel)",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.11.5"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}