```
{
 "nbformat": 4,
 "nbformat_minor": 0,
 "metadata": {
  "colab": {
   "provenance": []
  },
  "kernelspec": {
   "name": "python3",
   "display_name": "Python 3"
  },
  "language_info": {
   "name": "python"
  }
 },
 "cells": [
  {
   "cell_type": "code",
   "source": [
    "pip install contractions"
   ],
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "v3T25kTasJ1G",
    "outputId": "80246e1f-8e59-4530-b0ca-1489b00c0fb3"
   },
   "execution_count": null,
   "outputs": [
    {
```

```
        "output_type": "stream",
        "name": "stdout",
        "text": [
         "Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",
         "Collecting contractions\n",
         "  Downloading contractions-0.1.72-py2.py3-none-any.whl (8.3 kB)\n",
         "Collecting textsearch>=0.0.21\n",
         "  Downloading textsearch-0.0.24-py2.py3-none-any.whl (7.6 kB)\n",
         "Collecting pyahocorasick\n",
         "  Downloading pyahocorasick-1.4.4-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (106 kB)\n",
         "\u001b[K     |████████████████████████████████| 106 kB 5.2 MB/s \n",
         "\u001b[?25hCollecting anyascii\n",
         "  Downloading anyascii-0.3.1-py3-none-any.whl (287 kB)\n",
         "\u001b[K     |████████████████████████████████| 287 kB 37.4 MB/s \n",
         "\u001b[?25hInstalling collected packages: pyahocorasick, anyascii, textsearch, contractions\n",
         "Successfully installed anyascii-0.3.1 contractions-0.1.72 pyahocorasick-1.4.4 textsearch-0.0.24\n"
        ]
       }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "pip install pad_sequences"
     ],
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/"
```

```
      },
      "id": "utR_Lcd_r-x4",
      "outputId": "c90367ba-9995-4fcb-a43e-3eef8c14bdae"
    },
    "execution_count": 61,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",
          "Requirement already satisfied: pad_sequences in /usr/local/lib/python3.7/dist-packages (0.6.1)\n"
        ]
      }
    ]
  },
  {
    "cell_type": "code",
    "execution_count": 62,
    "metadata": {
      "id": "dECFcfw8rcYR"
    },
    "outputs": [],
    "source": [
      "import pandas as pd\n",
      "import numpy as np\n",
      "import re\n",
      "import collections\n",
      "import contractions\n",
      "import seaborn as sns\n",
```

```
   "import matplotlib.pyplot as plt\n",
   "plt.style.use('dark_background')\n",
   "import nltk\n",
   "from nltk.stem import WordNetLemmatizer\n",
   "from nltk.corpus import stopwords\n",
   "import warnings\n",
   "warnings.simplefilter(action='ignore', category=Warning)\n",
   "import keras\n",
   "from keras.layers import Dense, Embedding, LSTM, Dropout\n",
   "from keras.models import Sequential\n",
   "from keras.preprocessing.text import Tokenizer\n",
   "import pad_sequences\n",
   "import pickle"
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "df = pd.read_csv(\"/content/drive/MyDrive/spam.csv\", encoding='latin-1')\n",
   "df.head()"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 206
   },
   "id": "77BHAtE0swir",
   "outputId": "c5670d6f-cdf7-4c07-cd8f-3908bab2cd6c"
  },
  "execution_count": 13,
  "outputs": [
```

```
{
 "output_type": "execute_result",
 "data": {
  "text/plain": [
   "     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  "
  ],
  "text/html": [
   "\n",
   "  <div id=\"df-6baa4e01-52b0-4fcf-bca0-c1018c77c7f1\">\n",
   "    <div class=\"colab-df-container\">\n",
   "      <div>\n",
   "<style scoped>\n",
   "    .dataframe tbody tr th:only-of-type {\n",
   "        vertical-align: middle;\n",
   "    }\n",
   "\n",
   "    .dataframe tbody tr th {\n",
   "        vertical-align: top;\n",
   "    }\n",
```

```
"\n",
"    .dataframe thead th {\n",
"        text-align: right;\n",
"    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",
"      <th></th>\n",
"      <th>v1</th>\n",
"      <th>v2</th>\n",
"      <th>Unnamed: 2</th>\n",
"      <th>Unnamed: 3</th>\n",
"      <th>Unnamed: 4</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>ham</td>\n",
"      <td>Go until jurong point, crazy.. Available only ...</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>ham</td>\n",
"      <td>Ok lar... Joking wif u oni...</td>\n",
"      <td>NaN</td>\n",
"      <td>NaN</td>\n",
```

```
"    <td>NaN</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>2</th>\n",
"    <td>spam</td>\n",
"    <td>Free entry in 2 a wkly comp to win FA Cup fina...</td>\n",
"    <td>NaN</td>\n",
"    <td>NaN</td>\n",
"    <td>NaN</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>3</th>\n",
"    <td>ham</td>\n",
"    <td>U dun say so early hor... U c already then say...</td>\n",
"    <td>NaN</td>\n",
"    <td>NaN</td>\n",
"    <td>NaN</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>4</th>\n",
"    <td>ham</td>\n",
"    <td>Nah I don't think he goes to usf, he lives aro...</td>\n",
"    <td>NaN</td>\n",
"    <td>NaN</td>\n",
"    <td>NaN</td>\n",
"  </tr>\n",
" </tbody>\n",
"</table>\n",
"</div>\n",
"    <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-6baa4e01-52b0-4fcf-bca0-c1018c77c7f1')\"\n",
```

```
"            title=\"Convert this dataframe to an interactive table.\"\n",
"            style=\"display:none;\">\n",
"       \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\"viewBox=\"0 0 24 24\"\n",
"      width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
"    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.9.59 1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
"      </button>\n",
"      \n",
"  <style>\n",
"    .colab-df-container {\n",
"      display:flex;\n",
"      flex-wrap:wrap;\n",
"      gap: 12px;\n",
"    }\n",
"\n",
"    .colab-df-convert {\n",
"      background-color: #E8F0FE;\n",
"      border: none;\n",
"      border-radius: 50%;\n",
"      cursor: pointer;\n",
"      display: none;\n",
"      fill: #1967D2;\n",
"      height: 32px;\n",
"      padding: 0 0 0 0;\n",
"      width: 32px;\n",
"    }\n",
```

```
"\n",
"    .colab-df-convert:hover {\n",
"      background-color: #E2EBFA;\n",
"      box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
"      fill: #174EA6;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert {\n",
"      background-color: #3B4455;\n",
"      fill: #D2E3FC;\n",
"    }\n",
"\n",
"    [theme=dark] .colab-df-convert:hover {\n",
"      background-color: #434B5C;\n",
"      box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"      filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"      fill: #FFFFFF;\n",
"    }\n",
"  </style>\n",
"\n",
"      <script>\n",
"        const buttonEl =\n",
"          document.querySelector('#df-6baa4e01-52b0-4fcf-bca0-c1018c77c7f1 button.colab-df-convert');\n",
"        buttonEl.style.display =\n",
"          google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"        async function convertToInteractive(key) {\n",
"          const element = document.querySelector('#df-6baa4e01-52b0-4fcf-bca0-c1018c77c7f1');\n",
"          const dataTable =\n",
```

```
          "           await google.colab.kernel.invokeFunction('convertToInteractive',\n",
          "                                                    [key], {});\n",
          "         if (!dataTable) return;\n",
          "\n",
          "         const docLinkHtml = 'Like what you see? Visit the ' +\n",
          "          '<a target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
          "          + ' to learn more about interactive tables.';\n",
          "         element.innerHTML = '';\n",
          "         dataTable['output_type'] = 'display_data';\n",
          "         await google.colab.output.renderOutput(dataTable, element);\n",
          "         const docLink = document.createElement('div');\n",
          "         docLink.innerHTML = docLinkHtml;\n",
          "         element.appendChild(docLink);\n",
          "       }\n",
          "     </script>\n",
          "   </div>\n",
          "  </div>\n",
          "  "
        ]
       },
       "metadata": {},
       "execution_count": 13
      }
     ]
    },
    {
     "cell_type": "code",
     "source": [
      "df.shape"
     ],
```

```
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "CeNjrb-LyFsz",
   "outputId": "d0afbac8-e741-48f9-b4de-890b1596d3de"
  },
  "execution_count": 15,
  "outputs": [
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "(5572, 5)"
     ]
    },
    "metadata": {},
    "execution_count": 15
   }
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "df.drop([\"Unnamed: 2\", \"Unnamed: 3\", \"Unnamed: 4\"], axis=1, inplace=True)"
  ],
  "metadata": {
   "id": "C7EM1PHLyND7"
  },
  "execution_count": 16,
  "outputs": []
```

```
},
{
 "cell_type": "code",
 "source": [
  "df.columns = [\"SpamHam\",\"Tweet\"]"
 ],
 "metadata": {
  "id": "L5-AN1nmyS1U"
 },
 "execution_count": 17,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "sns.countplot(df[\"SpamHam\"])"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 297
  },
  "id": "zCNO_ZKJyYv0",
  "outputId": "5ea93418-3c2a-4833-9b02-a5214f24a6aa"
 },
 "execution_count": 19,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
```

"<matplotlib.axes._subplots.AxesSubplot at 0x7f9ecd5bbd90>"
]
},
"metadata": {},
"execution_count": 19
},
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 432x288 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYsAAAEHCAYAAABfkmooAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzA
AALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsIGh0dHA6
Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAYNklEQVR4nO3dbVBU993/8Y/caUyiS8SIAgM2wdaYGu8WdJL
WxvyDaGwwGWtJaUBDIdPopOl0WhmnGWy0U5mkpSbGzEipopMpMxMgajUPBFcFcFBdGyOOjjARj
DqgoIOVOD8H3HhlLzXqj6t6WJT3a+aM7GHP2e86O/ueszeHfpIsAQBwAwAA0PsRCwCAEbEAABgR
CwCAEbEAABgRCwCAUZCdO29sbNSXX36XX36pzs5OdXR0yOl0KjQ0VBs2bFMITIyOHj2quXPnqq2tTZK0YsU
KzZw5U+fOndO8efNUVVUlSUpNTdXvf/97SdKyZcu0bt26eG97uiRMndOzYMTvvGgDccaKjo3X//fW
XUtjY6M1ZMiQK9bl5ORYiRZixYtsiRZixYtspYvX25JsmbMmGFt2bLFkmTFx8dbFRUVliQrNDTUOnLkiBUaGm
o5HA7ryJEjlsPhuOHtulwu2+4TCsLy5263Oi5s8dfhkpKSlJBQYEkqaCgQLNnz/at/+aIobKyUg6hQ+Hh4Z
o+fbrKy8vl9Xrv1tam8vJyTm9vTYANCn2RoLy7JUVlamPXv2KCMjQ5I0bNgwtbS0Wlo0bNgwtbS0SJJaWlo0bNgwSVJE
RISamppp82zY3NysiIuK666+Ws/WkZEhl8sl8ulsLAwO+8WAPQ5tr5n8n8dhjj+nzzz/X0K0FDVV5ersOHD3/rOpZl3
ZLbysvLU15eniJ9XCkn0CAC6x9cji888/FhkpKSlJBQYEkqaCgQLNnz/at/+aIobKyUg6hQ+Hh4Z
o+fbrKy8vl9XrV1tam8vJyTm9vTYANCn2RoLy7JUVlamPXv2KCMjQ5I0bNgwtbS0SJJaWlo0bNgwSVJE
RISampp82zY3NysiIuK666+WkZEhl8sll8ulsLAwO+8WAPQ5tr5n8n8dhjj+nzzz/X0KFDVV5ersOHD3/rOpZl3
ZLbysvLU15eniJ9XCkn0CAC6x9cji888/lyR98cUX+vLDzX1XXF6fW1laFh4dkLksLDw3XixAlJksfjUVRUlG/
byMhIeTye664664664HAPQc22IxcOBA3XPPbb6fExISVFtbq5nEEqWlpkCk+Pl7t7e
1qaWlRaWmpEhISNGjQIDkcDiUkJKi0tNTuzQAAyFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IB
ADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBAD
AiFgAAI9tjERAQoH379mnz5s2SpJiYGFVUMjtdquwsFDBwcGSpJCQEBUWFsrtdquiokLR0dG+fWRlZcnt
duvw4cNKSEiwe2QAwFVsj8WvfvUrHTp5JydHubm5JydHubm5io2NdldfrVXp6uiQpPPT1dXq9XqXbsGxys3NVU5Oj
iRp9OjRSk5O1pgxaKtIFADIgDoDoSbY+60ZEROipp57S3N7N9+6adOmjamlKGdCg2bNnNS5K
SkpJUUFAgSSoqKtITTzzzhW19YYWKgLFy7o6NGjamhoUFxcnJcJ1jAwCuYmss/vrXv+rXv/61u3v/WDhgxRW
1ubOjs7Nzc7kIJUnQzNzc2KiIiQQdCksTU1NkqTOzk61t7frXdzyrBJAhV6y/epvKeR7cLnb7
qpp3TixAnt27fPrpu4SX9vrJtqbmvrFyA1gOBpUd00QjzLs7Xj8z1E2s+Kq1rYBztyF9tCwCqTrsFAPdpM7UxXbN0Pp5
iRp9OjRSk5O1pgxYY5SYmKhVq1Ypl
SkpJUUFAgSSoqKtITTzzzhW19YWKgLFy7o6NGjamhoUFxcnJcJ1jAwCuYmss/vrXv+rXv/udurq6JElDhgxRW
1ubOjs7Nzc2KiIiQdCksTU1NkqTOzk61t7frdryJAhV6y/epvLZWRkyOVyyeVyKSwszM67BQB9jm2xeO
qpp3TixAnt27fPrpu4Ql5enpxOp5xOp5xOp06ePWNkjtwkAfUWQXTt+9NFH9fTTT2vmzJkaMGCABg0apJW//kBUrVsj
hcCgwMFCdnZ2KjlyUx+ORJHk8HkVFRcnj8SgwMFChoaHq7u7Wq6+6Wk6+vr6emp6sWLVrl
qOTlZ27Zt089//nnNt375dc+bMUVFRkdRIS5S4wK7K2fFrS0Z/gVdAyCFgSUArt7TkFRj0Ca0aJqSFwqQjGAzGUF
u3XWMDAK7BtiOL61m0aJEKCwu1bNkyUSSSSlpaWpuLhYklRSUqKoAeUqK0tDRJ0pw5c7c7Rt2zbf+uTkZCIWEhCgmJkaxsbHavX
u3XWMDAK7BtiOL61m0aJEKCwu1bNkyVVVVKT8/X5KUn5+v9evXy+126/Tp00pOTpYk1dXVaePGjaqrq
1NHR4cWLFjgew8EANAz+kmy/D3EreZyuZSUlOSI0Om9qH6t277hF0+BO8lLcVH+PANjmRs+dfGEBAGBELAA

ARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARs
QCAGBELAAARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARsQCA
GBELAAARsQCAGBELAAARsQCAGBELAAARsQCAGBELAAARrbFon///qqsrFR1dbVqa2u1ZMkSSVJMTIw
qKirkdrtVWFio4OBgSVJISIgKCwvldrtVUVGh6Oho376ysrLkdrt1+PBhJSQk2DUyAOA6bIvF+fPnNW3aNI
0bN07jxo1TYmKi4uPjlZOTo9zcXMXGxsrr9So9PV2SlJ6eLq/Xq9jYWOXm5ionJ0eSNHr0aCUnJ2vMmDFK
TEzUqlWrFBDAAREA9CRbn3XPnj0rSQoODlZwcLAsy9K0adNUVFQkSSooKNDs2bMlSUlJSSooKJAkFRUV
6YknnvCtLyws1IULF3T06FE1NDQoLi7OzrEBAFexNRYBAQGqqqrSiRMnVF5eriNHjqitrU2dnZ2SpObmZk
VEREiSIiIi1NTUJEnq7OxUe3u7hgwZcsX6q7e5XEZGhlwul1wul8LCwuy8WwDQ59gai66uLo0fP16RkZGKi
4vT9773PdtuKy8vT06nU06nUydPnrTtdgCgL+qRF//b29u1fft2TZkyRQ6HQ4GBGbZKkyMhIeTweSZLH41F
UVJQkKTAwUIMHD9apU6euWH/1NgCAnmFbLMLCwjR48GBJ0oABA/Tkk0/q0KFD2r59u+bMmSNJSktL
U3FxsSSppKREaWlpkqQ5c+Zo27ZtvvXJyckKCQlRTEyMYmNjtXv3brvGBgBcQ5BdOx4+fLgKCgoUGBiogI
AAbdy4Uf/6179UV1enwsJCLVu2TFVVVcrPz5ck5efna/369369XK73Tp9+rSSk5MlSXV1ddq4caPq6urU0dGh
BQsWqKury66xAQDX0E+S5e8hbjWXy03lT+1ie8ctmgZ3kpfipvp7BMA2N3ru5AsLAAAjYgEAMCI
WAACjbsXi008/7dY6AMCd6Yafhurfv78GDhyoslAwORwO9evXT5I0aNCCga36LGgBwZ7phLF588UW98s
orGjFihPbu3euxLxZkZkZ7Ry5coeGRAA4H83jMWbb76pN998UwsXLiQOANCHdetLeStXrtTUVMUExOjoK
D/3WT9+vW2DQYA6D26FYt169bpgQceUHV1te+MsZZlEQsA6CO6FYt169bpbgpQceUHV1te+MsZZlEQsA6CO6FYtJkybpoYcesnsWAEAv1a2PztbW1
io8PNzuWQAAvVS3jizuuece1dXVaffu3V/5He45c+bYNhgAoPfoVizWrl1r8xgAgN6sW7H45JNP7J4DANCL
dVvFuxOHPmjCzr0opnMQ0JCFBwcrLNz/r+u+1f+CfFRcS3OHmjCzr0opnMQ0JCFBwcrLNz/r+u+1f+CfFRcSMEMCI
BerFuxOHPmjCzr0pnMQ0JCFBwcrLNnz/r+uBEA4M7WrVgMGjToistJSUVbpSUmaPHmyLQMBAHqf/+qss8XF
Zo+ffqtngUA0Et168jiqaee8v27f//+mjx5ss6dO2fbUACA3qXbsfhib1Pb53fszJkz9dChQ/rGGG/0DEF
XnjB7jkAAL1Yt96ziIIi0AcffABL1la1traqqKiv2cBAH1It2Kxdu1alpaWKioqSlFRUVq3b5/Wrl1r82gAgN6s
CX6FYshg4dqrVr16Z1O6q3s1OdnZ0qKCjQ0KFD7Z4NANBLdCsWBTk5KS0tTZZlyYoiZfbBQL97ar+mVmpdHQx48c1Z84czZ3z+bRAAC9RbcpS68psmI
4AEDv0K0ji7Fjx+7Jj+4Qq7HNaQWDcSXA+4x2vLK/ESDv+/Sb+3bGOAQ0/SvVgEBATI4XD4LoeGhl7/nPf9auXbv/vvvvS5J
+8pOf6I9//KOtgwEAeo9uxWeWcsAWl2+Vxs2ePpk2bJkl69tlndeqjQIVmEjzkHGahn0HhaY2HhsalLDdFjxDiU
IVsHAwD0Ht1+LenQqQa/Xyy9/9rKpw9XA3LlLlbE2+/HiLdfsOtsJQQPJOLbyFpIbTFRcdvWs/6NGzxQeI8jOw2dlbVVdWdyuT
gL6FWAAAjIgFAMDItlhZErZq27ZtOnjypUrr2tvVH+mrnRw/j7UnGGsUaJPo9hjdlTjknNzKjE8eqX9vKb3Yt/FJxiNtKsm1ENQRbdwHgJuT7
HKmpaqvr1d9fb1SU1PA9zP6b2xRVYdNWdmrKyckAVhFCGE2sHxaKkkfLFCbrab+Mynd+SJmpp6LQpMfk7ZMfuaWt7Hp5ecXFwzs7OcvulCAD72R4tLpYWVVVVVVSZK++uorHp0
SBEREUpKSlJBQYEkqaCgQQLNnz5YkjlSUlad26dZKkyspKORwOhYeHa/r06SovL5p8vbW+vmuSvKy8uVmJho1
9gAgGvoka9hR0dHa/z48aqrsrNSwYcPU/4Ii6VJunSMrnSH1hqamrybdcPcKyIIjrrr9aRkaGMjMzJUjbrJtrTQUUlhY
WF23h0A6HNsf4P77u1fvv1qZNm/TKYN//NTK6/oyy+//NbHL+B/dTK/vvOXWHvFwdO/AusTUWQUFB2
rRpk9599119+ciRYbdz1599119+ODHkqTW1laFh4c4XAPQcW5n9vQ4cOKT/T
c317eupKREaWlpkqQS0tDQVFxf71n/zSaf4+Hi1t7erpaVFWlSkhIkMPhkMPhUEJCgkplS+0cGwBwFdve
s3j00UeVmpqyqAwcO+N7Xrx4XxZSZYvX66NGZs0cyZM9P1KVJ6cyPJTxs0cyZM9XQ0KBz585p/vz5ki6d
Dn3p0qVyuVyLa0Da/Xa9fYYAIBr6Cfp1rxp0Iu4XC65Xv7P1R1S7eq5jgVdZNn8xUN9X+fb+4S/3BrRN/Gn6Ax2dFzJ9/Bg6AAY
EQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsA
gBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgBGx
AAAYEQsAgBGxAAAAYEQsAgBGxAAAAYEQsAgJFtscjqcjxxaemp860JDQ1VbXaCasWVZmj59uoYPH64hQ4bo888/18iRIyVJAwcO1LRp05SamqqqSgUFBRo8eLBSU1MlkGq
VlZXJ4X4D4XDC4SZrdPr1U9PmJPCLlS4uZl8z+d+FHOpp8wYcJTWp8Z5W5L06jRT2oXz0qsuVY5VmSu0zTGwcc0e1EBtl7bbWgJPaNrZZBvq+1Ld1wAAFQOnnp6BuQYnHIAK5vGTWEbplBSLekUC
aNUpbt25VlaVMbGJ05BfqYGzJBmKjY1V1ZLXTYTfAkTAVBSFmadZaZ4kmJKKSZqFMtMDJy7hy1Ih5z6ffcZIkvDxywnS7L0cLdyqN+SF0OEFfmzYS0L9aQpXhrGhwgo1H7Eo1A8eFpb1w2pBcP2fVkGbDOHJ4U4cXyASTlcsLP0WChUbeXcEsEjXDqTT3tcPk9F+1O6sZc0VHVbeRfjFZvqPvYy9rjLfVLLRLlTv2f1Sgwo0oEu2aEdrbLjwmlDHCr1TUOnDWfVZfAzSzEOnPyD3BzkaU8FmE0zOXtdXnJ7cmJW0cBLxHFUVuSqw4GdIBNRcvnpG98d4cakwfn8fRhWBlAmnEWw/vzAbn
2U/cBpyZCkoBeOlDBUBcC36zW0oIYx5xKMsPAMa0qkWVc3j7zVbGOxwoPHp1HiFv91V8yZfEBi2xxdjN/hzFO6tIE7tSfhkgVrWRbc9s6qZ2eXv1zZnb6AOO/bzYWsr6WXaGyW+QdKezXzXJWGvHsHsZa7ttlHoG/aIPtWEDDaaQqPYQKUqbsBDqCZa0KJFrJB/wGw4oPMb5AjdoLf2ocJ9HkC8kiP0d/P+hC+bzLL2Tv0nMefcHazZZqPndqtvOKBSrsDs0Gwyh/lYkK4jIuFmaPRMYcMh7gyB02+lXkcqu0ajIjJT8kAZ0j/m

1+HDh+V2u7Vo0SJ/jwMAfcptEYuAgAC9/fbbmjFjhh566CE999xzGj16tL/HAoA+47aIRVxcnBoaGtTY2KiL
Fy+qsLBQSUlJ/h4LAPqMIH8P0B0RERFqamryXW5ublZ8fPwV18nIyFBmZqYk6bvf/a5cLlePzngnCwsL08
mTJ/09Rq/QGx5XdQf9PUHvwWPzf92Kx2Z0dPR1f3dbxKI78vLylJeX5+8x7kgul0tOp9PfYwDfwmOz59w
WL0N5PB5FRUX5LkdGRsrj8fhxIgDoW26LWLhcLsXGxiomJkbBwcFKTk5WSUmJv8cCgD7jtngZqrOzUws
XLlRpaakCAwP197//XXV1df4eq89YvXq1v0cAronHZs/pJ8ny9xAAgN7ttngZCgDgX8QCAGBELPqw6Oho
1dTU+HsMALcBYgEAMCIWfVxgYKBWr16t2tpalZaWasCAAfrFL36h3bt3q7q6WkVFRbrrrrrskSWvWrNGq
Vau0a9cuHTTlyRFOnTlV+fr7q6uq0Zs0aP98T3O4GDhyojz76SNXV1aqpqdHcuXXPV2NionJwcHThwQJWVl
XrggQckSbNmzVJFRYX27dun8vJy3X//ZKk7OxsrV27Vjt37tTRo0f1zDPP+Lb/+OOPPPFRR0W3wAtNeyWPr
mEh0dbV28eNF65JFHLEnWhg0brJSUFOu+++7zXWfp0qXWwoULLLUnWmjVrrPfee8+SZD399NNWe3u7
9fDDD1v9+vWz9uzZZ49sPC8t/szz77LPW6tWrfZcHDRpkNTY2WosXL7YkWc8//7y1efNmS5LlcDh810PT
7feeOMNS5KVnZ1tffbZZ1ZQUJA1duxY6+zZs1ZiYqIlyfrggw+spKQkv9/P23XhyKKPa2xs1P79+yVJe/fuVU
xMjB5++++GHt3LlTBw4cUEpKisaMGeO7/ubNmRyvFxk7xTU1am1tkVsGDBxUTE+OPu4A7RE1NjZ588
kktX75cjz32mz/TpkyRdKlsziUlpbqwIED+u1pbpqwIED+u1vf3vFY/Tjjz9WR0eHampqcHZPPdn
x6j/z1i0cedP3/e93NnZ6eCgoK0du1azTz/7U3r17tXfvXt95kaZO/nq2bLFj/sZQC7Ha9EQmpeBsfS9/
mi999/X16vV9u2bdPIkSP9PP2dj9N9AOi1GhsbnAWtWn/9+AOi1GhsbNWnSJJ06dcrfo/R5vAwAYAIYAMSRBQDAiFgAAIyI
BQDAiFgAV1m8eLFqa2u1f/9+VVVVKS4uzrbb2r59ZOnOi7zJmA0VvxPQvgMpMnT9asWbM0YcIEXbh
wQUOGDFFISIi/xwL8jiML4DLhw4frzJkzOnv2rCTpzJkzvv/NERsbq5KSEpWXl+vDDz9URESEJkyYo
nap///+c/6kjR47t3s5+sr/nN7r4kHAOrpvo6+XnVTgUuNcLiOBH4TSVbZnZ4fx4u1RA3wn4Klq8LWLF
PBwKyQkxGpubraKiorU0NCgbgBfgFaKIi5xBL1xNthU0a5d3Dvvlh6+7XzvjNkCfYbLuvHChzvNAbsx4qj+5WyjDMGyGjaY+pXvc2eWjJT
5bZ2vWPB/P7W29Czgt0pP1Ycc+ePZC4vY4uz5s/JKD1KdalZU+DrBvlu0ccnZxHnX5jqTwkS8bvtrDr4
tvFFXNPy7FPA1Nx+p6o3c6h/4MGMncaAWrHdbT+7CT24ogE+nVkzCbmQadmVdO8hhp2pTfmhLUzNlHNZJFwSgSpNn0VRr/0nWnV9d6jF+
/zhDy1Jtp39dPv27dbhw4etqqoqq6qjyjp48KBVU1NjSbLuuusuq3///pYk6yfjq2ZyA0Gd3/1d8jTaKrq5Q+BCmlWoutKUjMljKlzjacg1ECgrWAJAgKqEmTSWLyJCOFq9KoLpSgbY5FF1MEB4vYpYZdubDfdQWYFqOc4RFb2BgcPLcg2F9eHbhFbnUbFH2M7CQmFxx8yFnxI
PBwKyQkxGpubraKiorU0NCgbgBfgFaKIi5xBL1xNthU0a5d3Dvvlh6+7XzvjNkCfYbLuvHChzvNAbsx4qj+5WyjDMGyGjaY+pXvc2eWjJT
W5urqRLZz/dsGGDhg8frpCQEDU2Nvr21d2zn6akpGjv3r2SLh1NfPTRR5Kk4OBgrVy5UuPGjVNnZ6dGjR
rl28blcqmlpUWSdOTIEZWVlfn2/fjjj9vxX4M+jlgAV+nq6tKOOHTu0Y8cO1dTU+E5Ud72zn/7IL3/R5s2bNX
XqVC1ZssR3nZs9++mvf/1rtba26pFHHHlFAQIC+/vrrb+37m/1dfjZgzqwKO/CeBXCZUaNG6cEHH/RdHjdun
I4dOyap+OnjwY/fyWZen5558nAvArHn3AZe655x3w/AZe655x699dZbcjgc6ujoUENDgzIzMVr1qweP/vpqlW
rtGnTJqWmpuqTTz7RV199dcv2ev6GAgAYcWQBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAo2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAAI2IBADAiFgAA
BGxAIAYPT/AXcn6tUKAJkyAAAAAElFTkSuQmCC\n"

        },
        "metadata": {}
       }
      ]
     },
     {
      "cell_type": "code",
      "source": [
       "def word_count_plot(df):\n",
       "   # finding words along with count\n",
       "   word_counter = collections.Counter([word for sentence in df for word in sentence.split()])\n",

"   most_count = word_counter.most_common(30) # 30 most common words\n",
"   # sorted data frame\n",
"   most_count = pd.DataFrame(most_count, columns=[\"Word\", \"Count\"]).sort_values(by=\"Count\")\n",
"   most_count.plot.barh(x = \"Word\", y = \"Count\", color=\"green\", figsize=(10, 15))\n",
"word_count_plot(df[\"Tweet\"])"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 863
},
"id": "NN-x4MunyhEM",
"outputId": "0e60b686-2ba1-45f3-b937-87c780869f99"
},
"execution_count": 22,
"outputs": [
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 720x1080 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAmkAAANOCAYAAABKr5voAAAABHNCSVQICAgIfAhkiAAAAAlwSFlz
AAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsIGh0dHA
6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzdf5DV9WHv/9eaVQGhqNmkSmNAiGNEE0E5i
QhIjXQmubWJVq2pxN93udFcf2Q01uuk4+bGdGhsk+baOKxxlhyvYar3KvpVa9VAlsUOeICi2BSUy4ZF38
MCslWkAjs949MafPFGKuc83nv8njMZGaXPXv2tZN/nr7Pfs6nLclAAAAoyj5VDwAAYHciDQCgQQCINAKBA
Ig0AoEAiDQCgQO1VD2iGl156KevXr696BgDAbzR27Ni973v3e3fh2SkrV+/PrVareoZZAC/UaPReMN/93
InAECBRBoAQIFEGgBAgUQaAECBRBoAQIGG5NWdGZOkq+oRAMCg1VX1ACdpAABFEmkAAAUSaQAA
BSou0kaPHp1LLrmk6hkAAJUqLtIOPPDMLVXrppVXPAACoVHFXd86ZOzbjx4/PypUr8+STTyZJHp06dNVW
OSGG27I97///YoXAgA0X3Enaddff33WrFmTSZMm5ckzn8zSSafn1PrufcVn6Pmnz9+cdOlOn2qThs01w
zM41GI41GIx0jOlq8GgBgzyou0v6t6dOn56677s7J7bsfONrenRu6vl8/uMO2eOy0KF
AnlV0pAEA7K2Ki7T+/v6MGj1+syzTzo6OnLSSdl2bJlFS8EAGi+4ceOWVV7JkyZLk09+Uq4b
mgQceyKpVq7Jy5coMDAzkmmuuyYsvvliJ1RACApisu0pJk9uzZvZV/L5NddcU9ESAIBqFPdyJwAAhZ6kvWM

bUsTd6wEA3i4naQAABRJpAAAFEmkAAAUSaQAABRJpAAAFEmkAAAUSaQAABRJpAAAFEmkAAAUSaQ
AABRJpAAAFEmkAAAUamjdYHxM3WAfgneuqegB7MydpAAAFEmkAAAUSaQAABRJpAAAFEmkAAAVq
eaR96UtfyhVXXLHr8xtuuCGXX355vvrVr6a3tzerVq3KH/3RHyVJZs6cmfvvv3/XY2+66aacf/75rZ4MANBy
LY+022+/Peedd16SpK2tLZ/+9Kfz3HPPZdKkSTn22GMza9as3HjjjTnkkEP+Xc/b+RqRqORRjhEdzZg
OANAyLX+ftPXr1+fll1/OpEmT8tu//dvp6enJ9On9OnTc9ddd2Xnzp156aWXsmjRotRqtfz85z9/y89br9dTr9eT
JI2+RrPmAwC0RCVvZnvbbbflggsuyCGHHJLbb789v/d7v/eGj9u9v+fGj9u+fXv22edfD/uGDRvWqokAAJWq5MKB
BQsW5OMf/3hqtVoeeeuihdHd35+yzz84+u8efueuih8e12S+PPC8BepqvqAewNGo1GarXdu6Xl+VY3ceddRRef
bZZ/PII480JdAAAIaClr/cuXbt2kyYMG5P2RD/NcPADCoueMAAECBRBoAQIFEGgBAgUQaECBRBoAQI
FEGgBAgUQaECBRBoAQIFEGgBAgUQaECBRBoAQIFEGgBAgUQaECB2qse0BRjknRVPQKAt6Sr6gFQ
JidpAAAFEmkAAAUadJHW399f9QQAgKYbdJEGALA3EGkAAAUaMld3dnZ2Zs6cOUmSjhEdFa8BAHhnhs
xJWr1eT61WS61Wy8YtG6ueAwDwjgyZSAMAGEpEGgBAgUQaECBBl2kjRo1quoJAABNN+giDQBgbzB
k3oLjV2yIG/YCAIOakzQAgAKJNACAAok0CiTQAgAKJNACAAok0CiTQAgAKJNACAAok0ACiTQA
gAKJNACAAok0CiTQAgAK1Vz2gKcYk6ap6BEDBuqoeAPwmTtIAIAAAAok0gAACiTSAAAKJNIAAAo0KCJt
wYIFefLJJ7N69ep0dnZWPQcAoOkGdanZWPQcAoOkGdxdWdF110UTZt2pRhw4al0WjjknnvuySuvvFL1
VqxYkaVLl+awwyCcsdtjOjs2G00mg0NsnWbMyg/8ME2ke/kM6h6EEaNNyIN5fGmAQAAATVd8pG344DGc5UWLsr4T7MzsfWX6PNtWWYEAm
mjVrMnfu3Xrl02kPYAATVf8y52/+8rMdHqnTG5kjrXfjuMJHPnvPU9r5KsqH2tCdPgWdabAhRbRkDt0
cN2e1y9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk+9Xk
BRJpAAAFaq96QFOMSdJV9QiAN9FV9QCgdE7SAAAKJNIAAAok0gAAClRZpl0ePTqXXHJkmTmzJm5//Jm7//
7q5oCAFCcyiLtwAMPAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAKFplV3fjwcAK
Z5Ikxx13XL7L72ta9l5MiR2bhxYy644IK88MIKLVc0GAGiJyk7Srr322vzkJz/J5MmT84Jf/fCGTJ0/OlVdemKTJ
2b8+PGZNm1a2tvbc9NNN+XMM8/c/9/rc9/W7cbSVc/ogI6QcZM2h9T7+YF0Tb4hS5+XUD8DcZfqmcr8j0Bv2+ZS
5alr68vSbJixYqMGzcumzumdvzvzjHHHJOHH344SfKud70rzz///OFrzf6tIKt9eT9uJ91z0JI0+hqtGQ0ACTFRNq2bdt2f
bxjx460t7enra0tTz/9HBmsKdhDHu/AK8EAttF5lL3f29q/f2Hfb/1Q12f3T9Qj+OqSW3xOPFT/kFqV8dmHvJPTsRheErgzAlN5aNqFlvIAE
ClKjtJe+WVV7JkyZLk2/VzXb09vnoeX/+B/Dv7Lf07WJxtrd5qc5K+gNKzoTsIgA9B/9ldDwQhzUHxjNLoeJxz8hFS4SL2HGSPiSGa91Ri8/P9H
mSg6hF7WqOvkVq9VvUMGF+vq+oBQCkajUZtd27xR0HAAAKVMyFA3vuhivVABgUHOSBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUKChetr3YdPEXA3vUhrvVABgUHOSBgBQoEEVaf39/VVPAABoukEXaQAAeyORBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUCBRBoAQIFEGgBAgUQaAECBRBoAQIFEGgBAgUQaAECBRBoAQIF
9+eQ4++OCMHDkyS5cuNxGn6/Uwc5SczS6ZR5K7r7vDVvw+Wz/kL4rX6cOM/eJlXHjNuoDwksxyPKdQIPngSXN+MKLOfdnF6rDKYp6iCUZN6Thz4V2Gp3Yw
2eWq2WWq2WjVs2NvV3AABotpa+me3es3ujTMMTMza9asTJ06NVu3CTj4ri2VBAoYrQZkNtmM6AAhwu9g2Da9v1rk6X9ACVeq8s2PHjrS3S3/+us
gYGBVk4EAChCS0/So8zWVG+EbM9qhnVW2u2b0M04aCBpBPdgjOyvs0bEm8Gzv5N2t8Tj9h7LfZaKbZTsnjfZkZAOMBFaGmkPPvhg2tv
s2bNmsydOzdlLy5909dfccUVdt4d37nd1q0EgCgem1JhtzriY2+Rmr1WtUzYOjnoAwNDVa
DRSq+3eLd4nDQCgQC29C29cKBlNsR/+QMAg5qTNAACiTSAAAKJNIAAAo0KCJtwYIFefLJJ7N69ep0dnZWPQcAoOkGdanZWPQcAoOkGdCiTQAgAKJNACAAok0gAACiTSAAAKJNIAAAo0KCJtwYIFefLJJ7N69ep0dnZWPQcAoOkGdanZWPQcAoOkGdCiTQAgAKJNACAAok0gAACiTSAAAKJNIAAAo0KCJtwYIFefLJJ7N69ep0dnZWPQcAoOkGd
AAok0CiTQAgAKJNACAAok0gAACiTSAAAKJNIAAAo0KCJtwYIFefLJJ7N69ep0dnZWPQcAoOkGdanZWPQcAoOkGdAAok0CiTQA
I23JkiVVTwAAqFSRkTZt2rSqJwAAqSEO4EAT0BPTRqRI
310NmzYkCVLlmTatGlv+HJoZ2dn5syZ4ZkyTpGNHR6pkAAHtUkSdp/9ayZcvS19eXgYGBrFixIuPGjXvDx9Xr
9dRqtdRqtWzcsrG1aZjzcrG1IwEA9rDiI23btm27Pt6Cq6xY0fa24s//AMAeMeKjzQAgL2RSAMAKFCRx2OGjUqSbJPsASAMAKFCRrx2OGjUqSbJPsa0a
IsWrRo179fdtllVU0CAGgpJ2kAAAUq8iTtHduQpQpKvqEQAAb5+TNACAAok0ACiTSAAAKJNIAAAo0KCJtwYIFefLJJ7N69ep0dnZWPQcAoOkGdanZWPQcAoOkGdAIA
CiTQAgAKJNACAAok0gAACiTSAAAKJNIAAAo0KCJtwYIFefLJJ7N69ep0dnZWPQcAoOkGdCiTQAgAKJNACAAok0gAACiTSAAAKJNIAAAo0KCJtwYIFefLJJ7N69ep0dnZWPQcAoOkGdANzRusj4kbrENX1QMAeCecpAEAFEikAQA
USKQBABSo6EhbsmRJ1RMAACpRdKRNmzat6gkAAJUoOtL6+/urnWLVqUnp6e9Pb2RvTpnb2Zvr06ZRuvAw
BorkHxFhznnHNOHnroofzZn/1Z9tlnn4wwYMWK3x3R2dmbOnDlJko4RHa2eCACwRw2SKGs0Grn99tz7
7775n/9r/+VlStX7vaYer2eer2eJIy7dr6Lv8rmdxeR6bzx5fz+8+n+/9CuZzwj5Cs7BsXJgYb2NXL
L3//e/Piy++mNtuu233Zt33XZjjvuKonAQ01aBuvfN3f/d384UvfOGKFWqFVlbiKi+Ae5bIpeb9Ck6Egb
NWpUkuTOO/Mnr/Xv6bGl4zesHeQOUQPOB0tdJSgY/w0W7P6bhSbwZnLTIdbT+6vYKOi+KBR6C00yP0BHnqZM7+s1b+sNnGXbE+E8qbmQDQVTUdMNQRjM0GlRBAUUoXVo

QCgQCINAKBAIg0AoEAiDQCgQCINAKBAIg0AoEAiDQCgQCINAKBAIg0AoEAiDQCgQO1VD2iKMUm6qh
4BTdJV9QAAWsFJGgBAgUQaAECBBl2kLVmypOoJAABNN+gibdq0aVVPAABoukEXaf39/VVPAABouiFzd
WdnZ2fmzJmTJOkY0VHxGgCAd2bQnaT9OvV6PbVaLbVaLRu3bKx6DgDAOzJkIg0AYCgRaQAABRJpAAA
FGnSRNmrUqKonAAA03aCLNACAvcGQeQuOX7EhbkINAAxqTtIAAAok0gAACiTSAAAKJNIAAAok0gAACi
TSAAAKJNIAAAok0gAACiTSAAAKJNIAAAok0gAACiTSAAAKJNIAAArUXvWAphiTpKvqEfAWdVU9AIASO
UkDACiQSAMAKJBIAwAokEgDACAhQ8ZE2duzY9Pb27vr8qquuyvXXX1/hIgCA5is+0gAA9kZD5i04Ojs7M2
fOnCRJx4iOitcAALwzxZ+kbd++Pfvs868zhw0b9oaPq9frqdVqqdVq2bhlY6vmAQA0RfGR9uKLL+a9731vD
j744Oy333459dRRq54EANB0xb/cuX379vzzX//pfs2zZsvT19eZZ56pehIAQNMVH2lJctN+Wmm26q
egYAQMsU/3InAMDeaFCcpP27bYibVgMAg5qTNACAAok0AIACiTQAgAKJNACAAok0AIACiTQAgAKJNA
CAAok0AIACiTQAgAKJNACAAok0AIACiTQAgAKJNACAArVXPaApxiTpqnoEe62uqgcAMBQ4SQMAKJBIA
wAokEgDAChQEZF22WWXZc2aNZk3b17VUwAAilDEhQOXXnppZs2alb6+vt/42He9613ZsWs3WNHC1YBAFS
n8ki75ZZbMn78+DzwwAO54447MmPGjIwfPz5btmzJnDlz0tvDz0tvbm+uvvz4TJkzI+PHj89Of/jTnnHNO1bMB
AJqq8pc7L7L7nkkmzYsCEnn3xxxo77LG57rrrcuedd+563MSJEzNr1qxxfG2idnZ1pNBBppNBrpGN
HRqvkAAE1ReaT9W9OnT8/f/f/u3fJkkWLlyYd7/73/773a7+3Xq+nnVquelVqulVqtl45aNLdkLAN
AsRUXam3n11Verngmg3n11VerngAA0DJFRVp3d3dmz56d2Y2btyY/5c2ycstpbtyCEnn3xxo77LG57rrrcuedd+563MSJEzNr1qxwfG2idnZ1pNBppNBrpGN
HRqvkAAE1ReaT9W9OnT8/f/f/u3fJkkWLlyYd7/73Rk1alSS5L777strr732a7+3Xq+nVqulVqtl45aNLdkLAN
AsRUXam3n11VerngAA0DJFRVp3d3dmz56d2Y2btyY/v7+ilcBALRe5RcO/FtdXV25/ucbs3LlymzbmzZsiX
nn39+1ZMAACpRRKQvdfvjhuz4+/fTTd/fTd/v6l770pVbOAQCoXFEvdwAwIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUKCheY
P1MXGDdVqnq+oBAAxFTtIAAAok0gAACiTSAAAKJNIAAAok0gAAClRppI0dOzZr167167Nd77znfzoRz/KvHn
zcsopp+Qf/uEf8uMf/zi1Wi0//GP09HRkSRpa2vLP/7jP+76HBgqKr8JO0DH/hA/vIv/zIf/OAH88EPfjDn
nHNOpk+fnquvvjrXXXd5s2bl9mzZydJZs2alZUrV2bjxo27PU9nZ2cajUjUYajUY6Rog4AGBwqzzS1q1bl9Wr
V2dgYCBPP/10HnnkkSRJb29vvvX+ua7ftPPd2Z2cajUYajUY6Rog4AGBwqzzS1q1bl9Wr
V2dgYCBPP/10HnnkkSRJb29vxo0bl9tvvz3nnXdekuiy7Kd77Kd77znTd8nnq9nlqtllqtlo1bdo84AIDBpPJI27Z
t266Pd+7cuebnzTt3pr29Pc8991xfefHFHnzzyyfnIRz6SBx54Kqp AAAtU3mkvRW33XZb5r3vffniF7+YG2OeeSYf/vCHs1gtmTk/ZHkuzRCSAwAO54M+tMVs+33rbvnKHEtWR7E03f/2w8FJFFNm/1jH7F/nsQxp2dcAA0BvK1AvCRFAHvLNL/eZ+TFmOKzuRHv1FhbWjo+MPZbmqjRjdK+TRmpaaHkq+oPWQKqlwbQOgnFVBtEbLQNvo1Kg8ZsFAdEldl+zzyUzMHo04HrkSH6P1bDrqqQ0xYIwU9Xc7rdvOkVjQwcOaMMw8GThsk6XgOa7Vd7BacE2Lv6HPBx8mN/ZtknjbdAIY8qAoEiVPmTDc3Pm+eSZZkPEW2XaVCo3yMbJdqO+x+AKSJtbbzs0s4k1ckQZ
R5Z4B7nMbHDBxhgCAAJImQehyNqkLbPR+dHYJUbBbQm8DPlpyiWpAZdr1Wn9+DGaAPCA7RKzNx1KufZ/o8z0N+AYF+dQZmdl4Et3QdX/9uM7B+JCMuNxhy9xz/E4aCp6vI8h8LUx/VHpfXEcfY1C2mzrv8uvHQpGD+8+NbrRuk7wR
CD4eFzKDxxMwFrHMTY4RoMt6L7UOwWwx3/TxQUyAtVQdrb3h7S6OBQK5Uz87EBqS3oLvZNHcRuLFZfyaWFb3gf5eCh/QvP3IYbf8+ViEfH6htAHGZOn2kjqKuN1RGBDFT3eYqoQTmuhTTq0WPw7Iz1qD9lgzTz5UI3i9wR/99
dhZKZiVgwdfvArK4IZr+Ct+g8JU7dEaDnhfCTDg3ayX6QdrmWoUqcuEg5FwvrlDg+PK0gfKJvQE/U8P6HvJzlxEzQyNlvNtCmqR/YLr2VoGLFnvUbBbGuowk4Wwdh+ZbE+sNOtzdYKi6rcRdiZiGZFxKU2gCw5AbGxsL3jqW0K
Yf6DfnlTYTD3qDiB4wK2b+wg/fN3yibr3xCdF9NZcIIzQsRmXjB3Fu9LoSmzr2WoEd2hKaggw0ZwAAAAAElFTkSuQmCC

Z3rNHXSK1eq3oGpeuqegAAJI1GI7Xa7t3ypn+T9hd/8RdJkj/8wz/MIYccknnz5iVJ/viP/zgvvvhiE2YCAJD8
hkhbvHhxkuQv//Ivf6XwfvCDH6TRKPiP8zfEKQkAMKi9pb9JO+CAA3L44Yfv+nzcuHE54IADmjYKAGBv95
beguPKK6/MD3/4w/zTP/1T2traMnbs2F3vSQYAwJ73GyOtra0to0ePzhFHHJEPfvCDSZJnnnkmv/jFL5o+
DgBgb/UbX+4cGBjINddck1/84hdZtWpVVq1aJdAAAJrsLf1N2t///d/d/nqquuyvve974cdNBBu/4HAEBzvK
W/STv77LOTJJ/73Od2/dvAwEAmTJjQnFUAAHu5txRp48ePb3t7bnkkkkty0kknJUl++M
Mf5tvf/na2b9/e1HEAAHurtxRpt9xyS/bdd9/cfPPNSZJzzz03t9xyzSzo7O5s6DgBgb/WWIq1Wq2XSpEm7
Pl+4cGFWrFjRtFEAAHu7t3R1544dO37l79IOP/zw7Nixo2mjAAD2dm96knbFFVksccey7XXXptHH30069
atS/LL20JddNFFLRkIALA3etNIe9/73pe/+qu/ylFHZV//Md/zCuvvJKFCxfmnnnvuyfPPP9+qjQAAe522JAO
/6UH77rtvpkyZkhNPPDFTp07N1KlTs3nz5hx99NEtmPjv1+hrpFavVT2DZuqqegAA7BmNRiO12u7d8pYu
HBg+fHh+67o0aOzYcOG9Pb27vGRAAD80ptG2re//e1Jq//e0nBvzA+Fve7vzxNPPTR6e/vzxNPPJHHHHnssX/va17J58+
ZW7QMA2Cu96dWd73//+5X//X8f+7nc1/cv+NPZ5a96q8wdmBBgDQAm96kvaJT3wiSXL00fnxVW1bSXL00UfnxBNPzFVXZVjjk
mr7zySh5//PF0dXW1YiMAwF7nLb1P2tNb53P93595Pb29vent7/+Tx544IEsWbIkEyZWbIkEyZMyBVXXLHHx3z+859Pb29vent7
c8UVV2Ts2LFZs2ZNbr311qxevToPoPPfRQhg0btsd/LgBAad400i677LLcddddWb9+fRYtWpRTTz01zzzzzTP7
wD/8wBx988B4dctxxx+XCCy/MRz/60Zxwwwgnp7OzMQQcdlCOOOOCLf/OY3c8wxx2Tz5s0544wz9ujPBQ
Ao0Zu+3Dlu3LjMz8/PCCy80dnOyc1/dcj06dOzYMGCbNmyJULy7733ZsaMGVm3rWMrjkn3bl1WrlyZJFm+fHnGj
3ht/f2dmZOXPmJEk6RnQ0dsSsAQLO9aaRdddVdVrdxa23ZsmFHfAOLOHHBxmAbChdGLcy7s3mHKG3Gnfm4-fHnGjRv
3ht/f2dmZOXPmJEk6RnQ0dsSsAQLO9aaRdddVVrdrxa23btm3Xxzt27Mjw4cPf8HH1ej31ej3JL98nDQBg
MHtLf5PWCt3d3TnttNMyfPjwjBgxIqeffnq6u7urngUAUIm39Ga2rdDT05M77rgjy5YtS5Lcdttt2bRpU8W
rAACq8ZZuCzXYuC3UXqCr6gEAsGf8uttCFfNyJwA0ZZgxDA3FCgL9v&gIADOGnCeHAACITSAAKJNIAAAok0
gAACiTSAAAKJNIAAAok0gAACiTSAAAKJNIAAAok0gAACiTSAAAKJNIAAAok0gAACtRe9YCmGJOkq+oR7
HFdVQ8AgNZxkgYAUCCRBgBQIJEGAFCgpkXa2LFj09vb26ynBwUqKmR9q53vSu33npvSu33nprVq9e
nYceeijDhg3Lf/yP/yP/yP/zHLli3LhUr8j//5//M8OHD81ut/uU/9Vv5//VHt0/9Vv5f//v/6WtrS1JMmEEiPz9pz9Ne3t3t7x8fnwceeCB
PPvlkFi9enCOPPPLKZkwEAitDUSDviiCPyzW9+H+M8ccc0w2b96cM6M844M844wz9ujPBQ
PznP8+KFSsyc+bMJMmpp56V2+dm2+++Q1/VmdnZxqNRhqNRj
pGdDTz1wIAaLqmvk/aunXLyiTJ8uXLM27ZcbcuB2Bb2bkyJF56KGHBHkiR33313zj777Pzwh
z/Mpz/96dx888054IADcuKJJ2Pm7nnP//fd/w59Vr9dTr9e9eTJI2+RjN/LQCApmtqpG3G3Xt27Mjw
4cNzxx135513/vnnOP/74P/PPPfPfPPPPFPxxcenJ9//MJ48uZkzAQCyenpm4pprPcqZ5evnDTR/dn3bp1OfPHM9c97sMf/nCr
K0/ILB0aNGpXnn38+n38+7e3tmT179q9bqz5TfjMfSKeLFQFsJsNf9/dGvgp/bO0tmT0d3fNxsgP
JwMAtFzJIr8P//RP88RaTI3TJkiV55enNfuVrdWt1d84999e/5ddhezuTlEqXmPw2dQAA
JwMAtFzLI+1P//RP88QTT2TJkiV55plnfuVrd999d84999d84999zcfffdu/5t9uzZufji7NixYo8/fTT+dSnPtXqyqQAA
LdeWZKDqEXtao6+RWr1W9W9Qz2tK6qBwDAntdoNFJraNbwtQFABjUnKQBABRIpAEAFEikAQAUSKQBABRIpAEAFEikAQAUSKQBANehcvwAACAASURBVBRIpAEAFEikAQAUSKQBABRIpAE
AFEikAQAUaGjeYH1M3GB9sOuqeqgegAAVMtJGBgAgUQaAECBRBoAQIGKj7T3ve99+efeTR/P0009n9erVuf
zyy6ueBDADQdMVFOLB9+/Zcdd/VVSnyciRIyV7N8+fI8oZa5NvyUzp4k0Vc5ZvGapj+53/O/vjpZgrf2OZvgk/+O4Y
2rVr8zu/8zsrVrwIa7Ij9L9Lpf2rbFjx2by5MUiSMK5vjJkb++okg0O4QU2PQsAokmKvgB2ZylBkQnEDEB
kyvrvDL9/f27fbler6dWq6VWq2Xjlo0VLAQA2HMGRaS8tn7nNy/7Dz6XfA8ebyZLs6u2ZSnR06kCdRrQ/7mb
7J27dp8/etr3odKAEBLFB9p9ZGZZ6o8KmKvvZZbD/ZUgR9ikklyvizwe/aGfl91ctxPKfjmqoOlkbHt5Grtr/qG3cjYPMtZyn3Crx+aoRo9O9kKi
FX+SBgCwNyr+JO2+D2MpKcJpTAEBBNYvrSC+3L5uMl2ZZbk0Ww5QmIbBHk4o2rAKV8SbpQGRqM+wHhmUfcvrDOAJPOXP3nyWBmk/+PzmyLIPcc/+cmPcQbseX0r+Wnv2l0gN
6MmzYsFbNBgCoRMsi7tdprr81PvKTTTV/4wwHLCAx+fadOmJUnN++q/IOh/5yEfyo
Q99KMOHD98pp+56ae+65J08+65WWMz+ZdyZMn57XXXtvt+Ts7OzdnZ3XVnt+Ts7O9NoNNJoNNJoqNVvxYAQFNUduHAs
mXL0tfXlw9/+MPp6urKxo9bH7u1Gq11Gq11Gq1bNyysYnLAQCar7I3s9
22bduuj3fs2JGtXs6b3r9fsbnhfgSn7Ht9vbsv//fmFvt/8/u/Lz1ev11Gq11Gq11bNyysYnLAQCar7I3s9
Geeeea/6/sBAIaKlp2kvfLKLbLKK1myZEjZEl6e3uzdzdevWWpii7s95mc/sv9/+1nq9XpWr16Q1Y3
K1q1bM3Xq1f8uzQAgKGiL2lcA1SP2tEZfZE6IlL9V6reoZ7WlVTdVmG2YAwDjREBLaAQQAIAICTdjBx1nYqkQAUZcYBRDcxeR0YHOAAKVNmFA021IU5q
AIBBzUkaAECBRBoAQIFEGgBAgRoAQIFEGgBAgRoAQIFEGgBAgRoAQIFEGgBAgRoAQIFEGgBAgRoAQIFEGgBAgRoAQIFEGgBAgRoAQIFEGgBAgRoAQ
IFEGgBAgUqAECB2qse0BRjknRVEPQL/HwDA2+ckDQCgQCINAKBAxUfaUz6eCgCQCINAKBAxUfaUz6eCgCQCINAKBAxUfaUz6eCgCQCINAKBAxUfaUz6eCgCQCINAKBAxUfaUz6eCgCQCINAKBAxUfaUz6eCgCQCINAKBAxUfaUz6eCgCQCIN
qPtGnTpiVJxo0a0bl3POOafiaJ2xydrDvdO3vQPdycfyd6/yvLhg8fY4/AAZZvB/JpEdbtjL8T93l6J8syq9XFnk+UnN3ZJBP++3mNG0HlbN3Gm/Cvo9xdae44y/vMVm+ssssasAJprOFzdee2110+2Bm+bhG6G9/OH/

zBH7zh1zs7OzNnzpwkSceIjlZOAwDY44o/SXur6vV6arVaarVaNm7ZWPUcAIB3ZMhEGgDAUDJoIq2/vz+
jRo2qegYAQEsMmkhbtWpVduzYkRUrVrhwAAAY8oq/cOBfTs+2b9+eU045peI1AACtMWhO0gAA9ibFn
6S9LRvi5t4AwKDmJA0AoEAiDQCgQCINAKBAIg0AoEAiDQCgQCINAKBAIg0AoEAiDQCgQCINAKBAIg0Ao
EAiDQCgQCINAKBAIg0AoEDtVQ9oijFJuqoegf8PAODtc5IGAFAgkQYAUCCRBgBQoKIjbZ99ip4HANA0lVb
QggUL8uSTT2b16tXp7OxMkvT39+cv/uIvsmLFikydOjWzZ8/OE088kZ6ennzrW98SbgDAXqHS4rnooosy
ZcqUTJkyJZdffnkOPvjgjBw5Mk888UQmTZqUl19+OWeffXamZ9+OWeffXamTZuWyZMnZ8eOHZk9e3aVkwEAWqLSt+
C4/PLLc/rppydBHZvn177rnnniTJKaecfkuOPPz6NRiNJMnz48Lz00ktv+FydnZ2ZM2dOkqRjEc
L1gMANE9lkTZz5szMmjUrU6dOzdatW7Nw4cIMGzYsr732vW9Wnbu3JkkaWtry3e/+91cd911v/H56vV66vV
6kqTR12jqdgCAZqvs5c7Ro0dn06bp1a4488ssiceIJuz3mkUceyZlnpn3vOc9SZKDDjoo73//+1s9FQ
Cg5SqLtAcffDDt7e1Zs2ZN5s6dm6VLl+72Hn744+72mLVr1+aLX/xi/u/u///b9ZuXJlHn744Rx66KEVrAUAaK22JANVj9j
TGn2N1Oq1qmfQVfUAACHfo9FIrbZ7t3g/CwCAAg3NG6xviFMcAGBQc5IGAFAgkQYAUCCRBgBQIJEGAF
AgkQYAUCCRBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFAgkQYAUCCRBgBQIJEGAFCg9qoHNMWYJF1Vj9jL
dFU9AACGFidpAAAFEmkAAAUSaQAABSo+0s4///zcd/NNSZLrr78+V111VcWLAACar/hIAwDYG1UhAeee
eee25WrlyZFStW5M4778ypp56apUn56M4778ypp56apUuX5qmnnsrDDz+c9773vvVNAwCoXCVvwTFx4sR88
45Bx10UAYGBnLCCSckSS6++OK9X66/3Pu3Tpg1s92Zhlm3Jp5mkzz///Xo+Jq7k2NvPz/Zc+HAWwPxvr7i/
O5z30uN910U84888888xMmTIlt99+e77yla+84+84fd2dnam0Wik0WikY0RHi5cDAOxZRb1P2k9/+tOCQ1rsZEJ
58+bluuuuyzHHHHJOH344SfKud70rzz///3///Bt+b71eT71eT5I0I0+hqtGQwA0CRFRdrAwMCvfN7f35+nn346J
554YkWLAACqUdTLnWPHjs0OJJ5yQJDnnnHOydnOnSvOc979n1b+3t7Zk4cWKVEwEAWqKoSHvmmvmmWfyu
c99LmvWrMlBBx206+/R/R/vzP/zwrVqzIihUrnKoBAHuFYl7uPAFY7uXL9+fY466qjd/n3YZDt7YsAAKpT1Ek
aAAC/VMxJ2h61J2kAAAUSaQAABRJpAAAFEmkAAAUSaQAABRJpAAAFEmkAAAUSaQAABRJpAAAFEmkAAAVqr3pAU4xJ0l0X1iCGuq+oBADC0OUkDACiQSAMAKNC
gibT/8l/+S9UTAABaZtBE2nXXXXVf1BACAlinywoEFCxbksMMOy7Bhw/Dhw9PT05Onn34
6n/nMZ6qeCADQVEVG2kUXXXZRNmzZl2LBhaTQamlzZv7zf/7PmTx58zsyZMydJ0jGio1o1VTAQC
aoshIu/zyy3P66acnSQ477LRv/F76acnSQ477LAcccQRv/F76vV66vV6kqTR12jqPgCAZisu0mbPnp1bGXJhh
w4ZVPQsKoKWKu3Bg9OjR2bRr0RpU7vFnNN5c8vxyE6n8FPrbXbyYybvxLTcnAyhvLVYtkG/y+cXE
mWjj8bodSs3ngnQk2XMP7dUHVpxkB1RJabh7cRTAeTdqbc4sAaPomA+AN8sx0KmlLYzNjkSMM7Uy8mVmpvKtIlmr
rJMJtk7uNJM0jdgrZxcOkdzC/SFAmbmRwBi0BAXrnbvSsjjKc0pgwqF0qEIAycBwzWx3i
exbOx4mrQcQUlb/VL0r7dF2FhY1LIT8jM6DhaOM4RQ8zZCw8NFp6pR9Jh+CXKIh7ObNsTuySzx
NDTxkzJMfjWsUaBWUWP6oMgXptrSdsmd2bRGNH2jmLJgm/ZgDnq5g1mTFSdmt4o5m4I1tmRIAQdn6

b++uCwcAAPZGbUkGqh6xpzX6GqnVa1XPGJq6qh4AAENLo9FIrbZ7txTxPmkAAPyqoflX+BvixAcAGNScp
AEAFEikAQAUSKQBABRIpAEAFEikAQAUSKQBABRIpAEAFEikAQAUSKQBABRIpAEAFEikAQAUSKQBABRI
pAEAFKi96gFNMSZJV9UjhpCuqgcAwN7HSRoAQIFEGgBAgQZNpJ1//vk59NBDq54BANASgybSLrjggowZ
M6bqGQAALVFZpI0dOzZr1qzJrbfemtWrV+ehhx7KsGHDcuyxx+bxxx/PypUrc+++bAAw/MGWeckSlT
puR73/teenp6MmzYsKpmAwC0RKUnaUcccUS++c1v5phjjsnmzZtzxhln5M4778yf/Mmf5Nhjj01vb2+uv
/763HPPPXnyyScze/bsTJ48OaNw78RC2sSu3bl1WrlyZJFm+fHkmTJiQAw88MIsXL06SfPe7381J
J530lp6rs7MzjUYjUYjHSM6mrYZAKAVKo20bdu27fp4x44dOfDAA9/2c9Xr9dRqdnRqtdRqtWzcsnFPzAMAqE
xRFw787Gc/y6ZNmzJ9+vQ4rZHlbRL
Mktxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

MS6devg5eWldjizTkTQ0dEBk8k04XVYpBEREdGMGxgYQHZ2ttphuBU+7iQiIiJyQSzSiIiIiFwQizQiIiIiF+QF
YOJ/C+omLly4gNbWVrXDmHUBAQHo6+tTO4xZx7w9j6fmzrw9j6fm7ml5h4aG4o477hjze6K1VlNTo3o
MzJt5M2/mzryZN3Nn3tNpfNxJRERE5IJYpBERERG5IB00+gFKdXV1aoegCubtWTw1b8Bzc2fensdTc/fUv
K+lyT8cICIiInJ3fNxJRERE5IJYpBERERG5IE0VaYmJiWhpaYHVakVmZqba4ThVcHAwKisr0dzcjKamJrz99ts
AgOzsbHR2dsJsNsNsNsNsNgMCjrfPDBB7BarWhpacGTTz6pVuhOYbfbYbFYYDabUVNTAwDQ6/UoLy9HW
1sbysvL4e/vryy/c+dOWK1WNDQ0ICoqSq2wp2Xx4sXKuJrNZvz111945513NDnmu3fvRk9PDxobG5W+
qYzvxo0b0dbWhra2NmzcuHFWc5iKsfLOzc3FyZMn0dDQgIMHD2LhwoUArv4fpb///lsZ9/z8fGWdFStW
wGKxwGq1YufOnbOex1m13O++PlXdJSYmSs91uh9lsBqCtMR/vGuYJx/l0qf5/QJzRvL29xWazSX
h4uMydO1fq6+slIiJC9bic1YKCgiQqKkoAiK+vr7S2tkpERIkZ2fL+++/P2r5r5iIgIqa+vFx8fHwkLCxObzSbe3t
6q5zHVZrfb5fbbb3fo2759u2RmZgoAyczMlG3btgkAMRgMcuTIEQEgK1eulOPHj6se/3Sbt7e3dHd3S0hIi
CbHPD4+XqKioqSxsXHK46vX66+er34+/tLe3u7+Pv7q57bZPN+4oknRKfTCQDZtm2bkndoaKjDct
e26upqWblypQCQl0eOyJo1a1TPbSq5fdlz/fsfz/lh5X9s+++wz+eijjzQ35uNdwzzhOJ9O08ydtNjYWNhsN
tjtdly+fBklJSVITk5WOyynOXnOXfunPLb1aVLl3Dy5EksWrRo3OWTk5NRUlKCoaEhdHR0wGazIcTY2drbCnRXJ
yckoLCwEABQWFuKZZ55R+ouKigAAA1dXV8Pf3R1BQkGpxOsPq1avR3t6O06dPjP7uMO06dPj7uMO4/5zz//jPPnzzzv0T
XZ8ExMTUVFRgYGBAQwODqKiogJr1qyZ3UQmaay8KyoqMDIyAgA4fvw4goODr7uNoKAg3DbbbbxzKJjDBn7iNoKAgLFiwANXV1Q
CAoqIi5b1yZWPlPp7x9m13PO/fLO/169ejuLj4httwtwxzEf7xrmCcf5dGimSFu0aBHOnDjvO7s7LxhEePr
QkNDERUVpRygb775Jhoa7b775JhoaGrB7927lVrHW3g8RQXl5OU6ou6cOIG0tDQAQGBgILq6ugBcnQSDAgIA
QGBgIM6dOwfg6gkgMDAw+2fHVWv4AsHnzZvzvvww/K6/DwcNTV1aGjgGgpxcEArr4fnZ2dyjLunvdk9m2tjXl8fD
x6enpgs9mUpPUi+XDm6lO56srCwMDw9zUTUUWi1UAZpYrz/fi6aefRmlpKQB4IdoMBAZXy/Px83ZPubiebl58eUjOGlC8T1cPj2ELmAfk9vh8jMCTXTCPnPIrRvTJRvK4+IjXrZ/Dm2Sgr2GMAyDbqIxJhFt6vjS9dz73mRcQEyRLNBBdY6Y9NUOa5eVyZK1ISUzIS8Zen4yeaIn6mwjIQqQ0LFdDcvJUk4Hr95s0hCVjRmY7FLAy9+ZSbG9iqUkg
x0mnG7fvl0AyFNPPeUw4bS6uEo4bfj1HKbTiouL5aWXXtL8mF9OnW7s0nN9j6eZ4z9kcJFWsiUZ/N5iB3j7OTyaBRWaDw0v1NMTpqHYZUW6ZaeBVA4pT2KiU+lTpfNE0VSFoU0Tnpq0gRlNXbWPUSsVhHlyjrOkyTd9xOZXbKR2RhXmBvGDyxYs+LI9qH6Flq4LJzN7wa
nNdm8ExMTpbm5WQICAhyWCwgIUMYzPDxcOjs7lfyun0RuMBhUz2squU9233bX8/5YfxCQmJgoVVVV
mh7zsa5hnnKcT6OpHoDTmsFgkNbWVrHZbJKVlaV6PM5sDz/8sIINDQ0iNlsFrPZLAaDQYqKisRisUhDQ4
N89913Die5rKwssdls0tLS4vJ/+XOjFh4eLvX19VJfXy9NTU3K2N52223y448/SltbmbMRUVDgcqLt27RKbzS
YWi0Wio6NVz2Gqbf78+dLX1ycLFixF+rQ45nv37pWzZ8/K0NCQnDlzRjZv3jyl8v8X355ZfFarWK1Wp1KGxd
tY2Vt9VqldOnTyvHeX5+vgCQlJQUaWpqErPZLW1tbJ27VplO9HR0dLY2Cg2m03y8vJUz2uquU9l33a38/
5YeQOQb775RrZs2eKwrJbGfLxmrCcc59Np/FgoIiIiIhekmTlpRERERFrCIo2IiiJjIBbFIlyIiInJBLNKIiiiIXBCLNC
IiIiIXxCKNiIiIyAWxSCMiIiJyQf8Hsv7ye0OcFyUAAAAASUVORK5CYII=\n"

},

"metadata": {}

}

]

},

{

"cell_type": "code",

"source": [

"pip install stopwords"

```
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "0wz_HSdFzoOz",
     "outputId": "2e9c6e98-3e09-4574-8ca5-272d7f8adfee"
    },
    "execution_count": 27,
    "outputs": [
     {
      "output_type": "stream",
      "name": "stdout",
      "text": [
       "Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/\n",
       "Collecting stopwords\n",
       "  Downloading stopwords-1.0.0-py2.py3-none-any.whl (37 kB)\n",
       "Installing collected packages: stopwords\n",
       "Successfully installed stopwords-1.0.0\n"
      ]
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "import nltk\n",
     "nltk.download('stopwords')\n",
     "nltk.download('wordnet')\n",
     "nltk.download('omw-1.4')"
```

```
],
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/"
 },
 "id": "V9HKQYz_0Se-",
 "outputId": "885c0101-dd27-422d-e6e4-329fc9a02f55"
},
"execution_count": 47,
"outputs": [
 {
  "output_type": "stream",
  "name": "stderr",
  "text": [
   "[nltk_data] Downloading package stopwords to /root/nltk_data...\n",
   "[nltk_data]   Package stopwords is already up-to-date!\n",
   "[nltk_data] Downloading package wordnet to /root/nltk_data...\n",
   "[nltk_data]   Package wordnet is already up-to-date!\n",
   "[nltk_data] Downloading package omw-1.4 to /root/nltk_data...\n"
  ]
 },
 {
  "output_type": "execute_result",
  "data": {
   "text/plain": [
    "True"
   ]
  },
  "metadata": {},
  "execution_count": 47
 }
```

```
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "from nltk.corpus import stopwords"
   ],
   "metadata": {
    "id": "CL5qwcfNzz4r"
   },
   "execution_count": 48,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "lem = WordNetLemmatizer()\n",
    "def preprocessing(df):\n",
    "    sms = contractions.fix(df) # converting shortened words to original (Eg:\"I'm\" to \"I am\")\n",
    "    sms = sms.lower() # lower casing the sms\n",
    "    sms = re.sub(r'https?://S+|www.S+', \"\", sms).strip() #removing url\n",
    "    sms = re.sub(\"[^a-z ]\", \"\", sms) # removing symbols and numbes\n",
    "    sms = sms.split() #splitting\n",
    "    # lemmatization and stopword removal\n",
    "    sms = [lem.lemmatize(word) for word in sms if not word in set(stopwords.words(\"english\"))]\n",
    "    sms = \" \".join(sms)\n",
    "    return sms\n",
    "X = df['Tweet'].apply(preprocessing)"
   ],
   "metadata": {
```

"id": "56wYerFPyr79"
},
"execution_count": 49,
"outputs": []
},
{
"cell_type": "code",
"source": [
"word_count_plot(X)"
],
"metadata": {
"colab": {
"base_uri": "https://localhost:8080/",
"height": 863
},
"id": "6SbWjKZg1Z9C",
"outputId": "278cb5f7-8352-4176-8d74-aa202b84b6db"
},
"execution_count": 50,
"outputs": [
{
"output_type": "display_data",
"data": {
"text/plain": [
"<Figure size 720x1080 with 1 Axes>"
],
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAn4AAANOCAYAAACP3QF5AAAABHNCSVQICAgIfAhkiAAAAAlwSFlz
AAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsIGh0dHA
6Ly9tYXRwbG90bGliLm9yZy+WH4yJAAAgAElEQVR4nOzde3RU9b338U9ikCQjFyW2wlITUYqGSxLIRoEE
CyJHMMglFC8pQqSTim28kSq19DCe1pZCW1tpATO0EZSLongAW8pFLonhhk0YMkMQ9EAOtYBohMRA
atSQ5w+fk+fhGBFtZnZmfu/XWq6Vmcze8/3x19vfnj2JktQkAAAARLxopwcAAABAaBB+AAAAhiD8AAAA
DEH4AQAAGILwAwAAMESM0wOEg/fee09HjhxxegwAAIAvlZiYqG984xst/o7wuwBHjhyRZVl09jQFH

lbNv+wt9xqRcAAMAQhB8AAIAhCD8AAABDEH4AAACGIPwAAAAMwV29F6KbJI/TQwAAgLDmcXoAdv
wAAACMQfgBAAAYgvADAAAwhHHhN3nyZM2bN0+SNGvWLE2fPt3hiQAAAELDuPADAAAwVcSE36RJk
1RRUaG9e/dqyZIlysrK0s6dO7Vnzx5t3LjxC/9YMQAAgCki4utckpOTNXPmTA0aNEgffPCBLr30UjU1Nem
mm26SJE2dOlWPPfaYCgoKLvicbrdbeXl5kqSE+ISgzA0AABBKERF+w4YN08qVK/XBBx9Ikk6dOqXevXvrx
RdfVNeuXXXxxxRerqqrqK53T6/XK6/VKkuyjdqvPDAAAEGoRc6n3f5s3b57+8Ic/qG/fvvr+97+v2NhYp0cC
AABwVESE3+bNm/Wd73xHl112mSTp0ksvVadOnXT06FFJn93JCwAAYLqIuNS5f/9+Jjl1ZZd4gGXve
8/nk8Xi0cuVKnTp1Sps3b9Y111zj9JgAAACOipLU5PQQbZ191JbltZweAwAwAAhDNPaN7Gtm1ZVsvdEhGXe
gEAAPDlIuJSb9AdU8gqHQAAIFjY8QMAADAE4QcAAGAIwg8AAGAIwg8AAGAIwg8AAGAIwg8AAMAQhB8AAIAhCD8AAABDEH4AAACGI
PwAAAAMQfgBAAAYgvADAAAwBOEHAABgCMIPAADAEIQfAACAIWKcHiAsdJPkcXoIADCUx+kBgMjBjh
8AAIAhCD8AAABDEH4AAACGMDb8xowZoxtuuMHpMQAAAELG2PAbO3asrkpOTnZ6DAAAgJCIiLi8SMTpjy1Ju6/pTOGoAbZVKSkaOXK00NNAMAAAA4LmI+41daWqrRo0erffv2/ruOALDkjsd/Vf9exVZAAAAALAFo8VMlkYF9SrRZHhu/S5hFHCgdNuiSJ1A/oJBJkFBRbW6ucnBxNnTpVe/fuVRtbDGSZ06hlOYVkHEeXlAcTvvTE+SwrKXRxaRmNPqoQojTjYEP6nWpHqWLpfDqSlXKXtrQCDRq/W7dauOc3Qef2YdRHZjJydhSJvhxegqTZfJpQiuoJEM
sU2CnL6NIiD6mUSSuBWhQ4wuGXbJkvS0aofmtwbsXvjjJG8yiHvw5mKqVjHvifKzhhDDp/K3tDZtBNRZaxQ5pmLMEcOOMeRkPC33jdM3xp3RnEoUz/sl9rzipQEmLv70SDlrrt/9ucf3Nsc6uXufIcWqs4hqUkuadMdBsBT4k+5eDdQEL7AyIQkIEtL5LXCKrAx7UoqKaHJgnDu1w4vOyuQRRUP2VbAqsBHJW19dQQOhqT4Ar2jYc/3eM9ZlEO3BpkVrmVU0jiMAXi7/sVe5+enpbDGMpWW1EbHnMUGZ+0eJpSUyLDUaUq40T97pWgNxaVPp9XxXMyP64n5s/Y9qyHYqSDbDRQlsywsa9RQDA5kU4aA8qHVDVaknF9jwqEVnx+ZuhHSCDHXqS0LkZJXtJG5xcvy6VEeGZYxhrGhyczc/mG0p1WPqVu4oExc3rEcHLOHKXAz3xCJ2L/kN8lD0Wua3DkkgWq2vDRWcyYp1Mmgj+QRWNAUZGOQZdM8sRZhGlYaVXGvZcTrrTyAL1yaXc0YGVlGDCDPTwSa+lnSG2bRj4Ui4bbLpq0OdUYVf3sG3xmMm6Z1GFbWf+PWl3wEXXC1VcMnCzZr0EWudwBJQ81qkeR63XzPOmMSMWxijd9ZgXx0dgUzp6NnFq6Yk6IsO1E2t3GyaAm+ZuqMb5/yvwyvDvvTTgrjWazeNjPtmBXGrBkN5MrHp4ZUhglZaxi63z7SAeovQ8vyYbmXSMRx9hI9CzhDMrQjRVeTkNw47IQDu6QWrKxAYkJGfdUMzbJp25tIU+E9xQcWdx1YzhNvJ7CRLqbCfPDS0PpyRlOMzeo/6YHwIAIMM23h3rb1ec5Kx2iX/VMwaVz1XmLOcZRgC0ALXy6HKo1NqJB+wzNWiJGXT5tLfTkszIKalRSfaMjCF8yM9K1hFbPbR9T9Wq/LPJhX+TDzkrlTmUQv9NTmOehXmFX1y9wHYF+6A0pbR6upb7W4DBlk/S+d1BvIXfyGPc6oYQXjpRHJOXcLYbSrLgfg4DpEOFKcejhOlRUbtW0E8tgWVAs8WwHRDcWm3fCj+r7wUUHUEe9QjuVX4cPpPrKXwzlDg9UZc9ZUAJFZT2jOa2pJkqq4rGMNrNpOx+pfqRDzJ4wX0/MEffsqVB6LHvCdGzlLgAUcScdlVKB7Bh9gW8gpNZ5Gv/OGtXMmzTp9qf8tShp97B5/3hrJsIPR8w+5VxkM5ps6RrLp6sRkBuCsdyFPiefnGoHNHW5lrWaM4b/FhCPjjyr4qsd0FXVOTdyJW1sWMNcG/zB0eTHO0OQ4GdB/rpNqNXjZ1IXnfBVWbk8+mnVezUfJV2A9+6oxdjC7HXozyYl1UsrrArgohHsApnERQbG9fD2uXzX9B6y/jr8K7Zrkn3g8BR10oABlwQ9g9mLo33XGaLcvNdHYBdtFcb55Wj9IcdCOOuP2snp8h+Hr4ULh6CFe7lJAgc1/u4Nhgh2MM1xf/iq5bU4kzHv3Ofn7F6q5g+kuYQ0eKJGeNUuBtAcN70Aqv8QSdZEUP/N5MVIEtGSZsWLjUSDC66MxSSDh8Csq7OtjyvmG9d2a6s8+mYwwGQ1h7xxbqs8BqzFJ/qbk5vnGZCdtIHFqcOZgtvq1fsiJnEdSrkuzvIOwUNkpdlvSQHhgmwcOG6vl9i3JXFunGH4xfDvV1Yfg6s1D6Tm4uCnYjbv6dfUyDc6o1NNJRPMPXHT4uJDSMI5ypJeFFZ5Xt9XqRpPPxSxIjMomwqgSGIPXlvyjH6dUJTc9ZdkJAXHjzDWn1bcFebvNDLQPBP5ItmWUEsz/zJFYoDeCeF2VyFBKlxeuFEqaxSGhdCeAoLmY+Hx0Fke3rCCOoVk1Z3SPMS2FZLDoCBS9eK6vYG6Nve6Cwl3ZeL+eTm5PHg/p7Fq9ELH8LaHpOWtijNJhVzbAQ65hBZgvsj9Ryb0SZLM9ZHQ9b+f7uu7FUa81vdBaGPo6fjGWEqrRBYLGhZm9+L1GSukGhw7HyBtzdr+Rqb0e4VWeyRs48GR9xyKW0aESlnqN0s4u+pHZXyPHsuZ6dc34jGeGGZGMRzpW1Pp1aHnlQ5IrkddNSebYIlPjGBEOyDwANNeXbahSUsSb7aCdeHjtBfVFJ6rVyxhKY1Cuh/pEvZKmNb8Bkn0ZIW5JA4ys9nyLPHsWz2S7ALGvbXeGOgI7ErtAq53BAAAs4A8LxlrY0O9J3OMoo8JnIv8chQ4jEx5S8BwHhSmgzEkzKbXkmhsv62gIfzbJc5wHNtf3Dk40FPxsV+XoP1cQwXkd8kMOL0F8v2OOYvHxAV3SDCxmm2J2ayJRa9RLCABaMzyIU5cRW0GnrcSMtH05AWJmA6L03Eo7WxOSqh5wtSVTkWWGoDPfdUjZ9r+XSVWbY+1qp8lnHvfRgTfKxtOXvbS7Ycs6VA97Fwvbdoip9MoUXC6v8Ja6fMc0MWJNWo5JjrE7B8ArwiY1tMnP+MB6pHPf3sK0DK4jWxQChK8YYXr05E/Wmrdkn5tbGYKnBmhsDRf36WG8L6JmTNUzcdBo8+3Qtl0ovwsmf3fPqLZo3XlVgX3XVdKKfPbkdHu+Mh59e5hUjEUB2t9NK4mR6vFzBqHjfiWRh9dBKWDfmOKGNxcwC98VZMZ05xpWrVGyPRUljDU8NCoKfdxp43klc0+O9L6+pP3a8mRuG5DAvvbKMRZwPOkf0aiyIvlWfEbtgvOHXJjv7ZHC2Obv5WshqDkOZmuKhE7zuC4D0GwjbSXUgB6EBozJ1ZhQpxhPcJt/2HtV4kpr2NBz11O+nRYjddNGoWMpoa8ESmTAFHlZsM28MEoDTswRhmGhsUIBIdBlRHbmJJHQeuU7uO8olcDSZXfsFFrzlIQuwQP8OY2PVcG1pR2xPWzJmqMNENdBtxLTmbc4TfOR3a/mxwkrsq8VN8gbdbL1w2GCqcOs2sdQe9LOybWRNnWxMY0UQpGUW5pxWsvkd1nNZiB9D3WWNxQKYAvJDgJx3L6eJYSmsyqy6pmK45OOOeGP4qWd05KvWzWoz4n/5wH4DRkS2pFn8iWrHa5BSZz8WkGCBsNMPZrKbgKw6mbBEiDJJkc/3cuYOpxdYR9w0Z4d9dFT2NIrKnF8CPe1h6RrRqW1U8nqs2JzcGNC3Amq0CTukSG8+Q8iFcxibqEXN+aB8BxMrMbjZ/C4nYPX5OsuQ5o7aN97yXYRNvAIhXfSBmeM5BoPvfEWuVEO3oEOsy58bOb/0BSh4BtQqWXZ4cvWrNjAA6C6dAZeNoHUVE1GvaTGGUQ5Ms8dQ6l7BD3JeqUsRc3QhGl4JlZMzFm88G2tTKvz7SOJ5qsK2T4B2OwRgZ+CDtw9WV4U/4A8sS05f1hLIaOb46tr4hRJfqqeNKH6dXg7oIN1tlKnNT0WjfSqI6tKUiwPhpZqZGuCOMy5oR1d0OT4spnQYAS4/TbEbIvyLxOVFo7MNPu6u6ubv5CNeC9yNMEK7bFUwWCsBoxC6e8PvWq1aGJ2m0N2vZOzqSwvnG1JsMYD7SF2rrnOWx6yZvHN4y7KXmtmd4LSVYtBS5KULWTU41ap5cNDE8aWKw0Kt2eS9nOaDUHgYn6Mg3V7w9

PwAAAAMQfgBAAAYIsbpAcJCN0kep4cA0Co8Tg8AAM5hxw8AAMAQhB8AAIAhCD8AAABDGBd+iYmJC
gQCTo8BAAAQcsaFHwAAgKkiPvweeeQRBQIBBQIBPfTQQ+f87pprrtGePXuUnp7u0HQAAAChE9Ff59Kv
Xz/l5ubqxhtvVFRUlHbt2qVt27ZJkr71rW9pxYoVmjJlivx+/+eOdbvdysvLkyQlxCeEdG4AAIBgiOjwy8jI0Ku
vvqr6+npJ0qpVq5SZmanLL79cq1ev1vjx4/Xmm2+2eKzX65XX65Uk2UftkM0MAAAQLBF/qbcltbW1+vvf
/66MjAynRwEAAAiZiA6/kpISjR07VnFxcYqPj9e4ceNUUlKijz/+WOPGjdO9996ru+++2+kxAQAAQiKiL/X6
fD4999xzKisrkyQtWrRIp06dkiTV19crKytLGzdu1OnTp7V27VonRwUAAAi6KElNTg/R1tlHbVley+kxALQG
j9MDAEBw2bYty2q5WyL6Ui8AAAD+n4i+1NtqjoldAgAAEPbY8QMAADAE4QcAAGAIwg8AAMAQhB8AAIAhCD8AAABDEH4AAACGIPwAAAAMQfgBAAAYgvADAAAwBOEHAABgCMIPAADAEIQfAACAIWKcHiA
sdJPkcXoIAP8yj9MDAICz2PEDAAAwBOEHAABgCMIPAADAEBETfnV1dZKkxMRE3X333Rd0zI9//ONgjgQ
AANCmREz4/Y+kpCTdc889F/TaJ554IsjTAAAAtB0RF36zZ89WZmamfD6fHn74YcXFxenFF19UZWWlVq1
apZ07d6p///765S9/qbi4OPl8Pr3wwgtOjw0AABB0Efd1LjNmzFBBQYFGjx4tSZo+fbpOnTqlXr16qVevXtq
7d6+kzy7z/vCHP1RaWlqL53G73crLy5MkJcQnhGZ4AACAIIq4Hb//LSMj/ZZW/3X9BxXq9X
mXJsixV11cHc0QAAICQiPjwAwAAwGciLvzq6urUoUOH5selaWaOGiJOmGG25Qnz59mn/3ySefKCY
m4q52AwAAtCjiws/v96uxsVF79+7Vww8/rPnz5+vyyy9XZWWlfv7zn6uyslK1tbWSpMLCQvn9fm7uAAA
ARoiS1OT0EMEUHR2tdu3aqaqaGhQd27d9emTZvUs2dPffLJJxd8DvuoLctrBXFKACHHcXoAAAg+27ZlWS1
3S8Rf54yPj9eWLVvUrl07RUVF6YEHHHvhK0QcABABApIj78Tp8+/YXVe8GOiZ0CAAAQ9iLuM34AAABoGeE
HAABgCMIPAADAEIQfAACAIQg/AAAAhiD8AAAAAhiD8AAAADEH4AQAAGILwAwAAMAThBwcMIPAADAEIQfAACAIQg/AAAAQxB+AAAAhiD8AAAAhoi48Js1a5T5/
u9BgAAABtTsSFHwAAAFoWEeH3xBNP6ODBgyopKVHPnj0Vzd8vSd/3tkqLq1at0qeffvq587rdbtm2LkKVWlamYvn36uWXWX1ZcXJwuueQSH
T58WDExn93M3KFDh3MeMwAARLKwD79rr71WY8aM0dChQzVp0iT9/PlCh9cU3nPcPHhhSAmbjhbQ9D4jebhIpCQr
T58WDExn93M3KFDh3MeMwAARLKwD79+/frprvvuUmpqqkaNGiRJRjZBATAAQF7D0mamqo333xxTU6d
O1enTp7V161bdfvvtkqSS77rpLq1at0qeffvq587rdbtm2Ldu2lRCfENI1AQAABEPYh19mFVV/fOf1R
dXZ3WrFkjSerd7u6eKi4vVqaSVVqk3NxcSVJubq0buaspPK/X65WkSbIsS9X11kk77Tuaq6x3Ij26IaVl5Ki/pK/OGkXZFDABIoebtTiSxq
F7DXO5557TmPHjpXf79fkyZP79fkyZP17W9/W5K0fft2JSUl6eabb9ZFF12kyqk6spKZwcFAAAIkbDf8SsuLtbYsWMV
GxurSy65RKNhj5b02ef3jh8fh8/rpiYGOXXk5JxzzJlJIS7Rs2bIv3O0DAACIRGEffj6fTTy+++KlqKiq0bt062bYtSfrrpT
3+qXbt2qbS0VAcOHDjnmKVVLl+rSSy/V8uLnRgZGZAADERFxqfV/iFfvGLX3zzu+YULF7b4cOL
qrm0N9mgAAABtRkSE31fxxWTSMNOaTlowapTTowAAAISUceH3PwAAZHm4Dp0RvAVDAA/vKmAAAAOUceH34IMPPFfvWDj/knytPYkAAAAAoRX2n/EDAAAAPisDAQjlEdIiYwPZYQmNZkz7SB4bmNK7ZhNumY6RqYTp8V4+GjFJUsSlZ3RkMpmLZ8
xcx4VdaWqrRo0erffv2/crclcysrKKK0pZE3Tq1ClZaGRIkiZNmqRt27Zz3ZJkurq6qgSd/QcnRwYAAAipiLm5Y/fu3vq
zZo38fr9rOnDiDZ/S4ZtpaTZ48ufnmjsOHDys3N1eS9MfK6mfs1jmMdsEqwmBJ3fdYr6EbnmxBWpNZzZx+6r9f4Hup5ihfNQ+n6jUYCyaYHnsWikLMFM43d+hHiAp0cBFGvgEwxP2xfrsWjOvy
i4mLl5eXJ5/P9y+e1j9qyvFYrAjggnicHgAwpdt27KsrXlvsrslYnb8JKmwsFDJycmKY3V4sWLWyX6AAAAlkV
EhV9OTk5wTxM7EAAAICwFzE3dwAAAOD8CD8AAABDEH4AAACGIPwAAAAMQfgBAAAYgvADAAAwBOEHAABgCMIPAADAEIQfAACAIQg/AAAAQxB+AAAAhiD8AAAAhoi
BOEHAABgCMIPAADAEIQfAACAIQg/AAAAQxB+AAAAhiD8AAAAADEH4AQAAGCLG6QHCQjdJHqeHAAzi
cXoAAIhM7PgBAAAYgvADAAAwBOEHAABgCMIPAADAEEwhLHhd/PNN2vgwIFOjwPbB/ZDBgyopKVHPnj0Vzd8vSd/3tkqLq1at0qeffvq587rdbtm2LkKv
oKBA+fn5kqTf/va3ev311xVJQ4cO1QsvvKD58+fLtm3t27dPPHo+n+biqqip5PB6Vl5fL7/erZ8+eSkxM1P33
369HHnlEPp9PGRkZTiwjZEgwJAAAgpMIm/EpKSpESPmEpKSmZSZSmsdqaz3/n1l1P/nJ72Raub27dn5zHLQaePZaRa/fpp+m/9h30Jdp9/1m1tR3nhqtGhm0opUyWaxbb5V5VkmzJ5oeUKHxbLh1Ty2
75Zff0aT62urpa/fv314IFC1RQUKAjR45R0PQUKAjR45o4cKFevfu7Fevrpp5WWWHqrpMfvKtDqWUBAACETNiEX3l5ufr27av58+enr3768qN1cOFiBUbSW
ooaFBO3bsUHp6ujJycjLxcPp9PvXr1Oxq1unrJycvOxq1ataj5HULLSBb2f2+WbduybVvbVsJ8QnBW
BIAAEBIhc33H366aer37dvn9vt1nXXXafHHntMO3bs0KxZs5PPnz/t+x3fg9Jf+3+S/nYrNXmcTrSHPmL0aUVkZEKJy1Iw7dU7ofFGhN4MDk5ZQUG
S1NjYqJiYC+1uV1+uV1JNlH7dZfEAAAQIiFzY6Y6f9rq9WsqSubrOxrFBr3r27fbTe8jDAG3aWAbqyeDPU2Zt3GraGGnd2+1NX+nWHnQsCyHsnBDWHtrVgEB1xkLz4gcHrR6b
+MY3NHLkyC89fvqpC89V11dnTp06sKNQtQrXkuBeb/vSPjF5XWdrcaNTzVwSyDPdNGHNdC2dz0aPtywb/1yMaFbhUsCQBMNCpmnZx3uwNR/9Bwfgz
Wlr6pedau3atxo0bx80dAPAGFGSmpweoq2z9qyvJbTYWdm8Dg9AACEL9u2ZVktd0YY7fgBAAAAAOAwBgz
bmzscdUzsQAAAgLDHjh8AAIAhCD8AAABDEH4AAACGIPwAAAAMQfgBAAAYgvADAAAwBOEHAABgCMIPAADAEIQfAACAIQg/AAAAQxB+AAAAhiD8AAAAhoi+ZklVVoFBF

MHAABgCMIPAADAEIQfAACAIQg/AAAAQxB+AAAAhmjz4Tdp0iRVVFRo7969WrJkiRITE/X666+roqJCm
zZt0lVXXSVJKioq0vz587Vjxw4dOnRIN998s/70pz9p//79Kioqaj7frbfequ3bt6u8vFwvvfSSXC6XU0sDAAA
IqTYdfsnJyZo5c6aGDRum1NRUPfTQQ5o3b54WL16slJQULV26VM8880zz6y+99FINHDhQjzzyiNasWaO
nn35avXr1Up8+fZSSkqIuXbpo5syZGj58uPr376/du3fr0UcfbfG93W63bNuWbdtKiE8I1ZIBAACCpk1/j9+
wYcO0cuVKffDBB5KkU6dOaeDAgRo/frwk6fnnn9ecOXOXOaX7927VpJUiAQ0IkTJ7Rv3z5JUmVlpZKSknTll
VcqOTlZpaWlkqSLL75YO3bsaPG9vV6vvF6vJMk+agdngQAAACHUpsPvq2poaJAknT17c+KKkWzt2bD0fqa7Vv0
Gbdy4Uffcc49TIwIAADimTV/q3bx5s77zne/osssuk/TTZpdzt27zfrrrvukiTl5OSopKTkgs+3cc+dODR48WNde
e60kKT4+Xj169Gj9wQEAANqgNr3jt3//fj/J/R6oRz/S+++/r6vvn
MmaLly5erffv2kqSZM2fq7bffDtYSAAA2owoSU1OD9HW2UdtWV7L6TGAyOZxegAAiAy2bcuyWu6W
Nn2pFwAAAK2nTV/qbTOOid0IAAAQ9tjxAwAAMAThBwAAYAjCDwAAwBCEHwAAgCEIPwAAAEMQfgA
AAIYg/AAAAAxB+AEAABiC8AMAADAE4QcAAGAIwg8AAMAQhB8AAIAhYpweICx0k+RxegggQnmcgA
AzMGOHwAAgCEIPwAAAEMQfgAAAIaImPCrq6uTJHHXt2lUrV66EJEUOmmsJUr7zciwAAIA2I2LC738
3/nOd5weAwAAAoM2JuxfEBQOzT4zrEMSeEgiw4EoK7GwIcuGlg+KaBZFSs1xTiY8PCCXAABiUWRRRby8pR
EOl35tPwuq2TlZgA+lA1B
HUAmR8y+5PmGwBfDtiNeXnMemU/TfTQfRvMcRdRovpBWSa4C13//VX4/IQW9zUSdm8LMv6bqw5xb/ZfIUSPpaGpTWoijEnVXq0EBh==
EjZ1RNb8UvVRCRBVGQdFk0mznb7FeTY4KSnlcTPP6OM8G3Xm8F52kamySi2nL0y3bgOyMTgQeQYANDLvmqzVRGBwy3zrC8fXFF6VSD
xHZFz2nMaVnBl6J8vnIP8eN3S8VDCuZRQyJNR1mYjjMKjXiZHnu7Zsnq+tZ5UDDIF6m
uGMfcGYwYaUjQH2NOWuJUNSJTpKBVoMgQrC0
AAAATRBVGQh2PgNf4qnMuz2ZKxl4FwgFKVYJnKWNyByenR0bJQW7iSjGV4

W1uryZMnN9/EcfjwYeXm5krSFz6fm5urP//5z2pqauLmDgAAYJQoSU1f9Ms1a9aoqekLf60xY8YEY6Yv5
HK5dObMGcXFxam4uFh5eXny+XxBf1/7qC3LawX9fYCI43F6AAAwj23bsqyWu+W8O36//vWvJUnjx4/X
FVdcoRdeeEGSdPfdd+vEiROtPOaXKywsVHJysmJjY7V48eKQRB8AAECkOO+O3/9oqRzPV5ORxqS1AgCA
8Ha+brmgmztcLpeuueaa5sdJSUlyuVytMx0AAABC4oJu7nj44Ye1detWHT58WFFRUpMTReXl6wZw
MAAEAr+tLwi4qKUqdOndSjRw9df/31kqQDBw7o448/DvpwAAAAaD1feqm3qaalJJz32mD7++GP5/X75/
X6iDwAAIAxd0Gf8Nm3apOnTp+vKK6/UpZde2vwfAAAAAwscFfcbvzjvvlCT94Ac/aH6uqalJJ1157bXCmAg
AAQKu7oPDr3r17sOcAAAABAkF1Q+MXExGjatGkaMmSIJGnr1q169tGr1q169tln9emnnnwZ1OAAAAALSeCwq/BQs
WqF27dpo/f74kadKkSVqwYIHcbndQhwMAAEDruaDwsyxLqampY+3bNmivXv3B3Bm0o0AAAAAtL4Luqu3s
bHxnM/5XXPNWpsbAzaUAAAAGh9593xe+ihh7R9+3bNmDFDmzdzdvVlVlVlaTP/mTbfffF5IBAQAA0Dr
OG35XXnmlfve73+mGG27Q22+/rZMnT2rLli165ZVXdPz48VDNCAAAgFYQJanpy17Url07aena9CgQRo
4cKAGDhyompoa9erVKwQjOs8+asvyWk6PAYQPj9MDAIC5bNuWZbXcLRd0c0cdcXJW6duyoTp06tVQUVVZWlgYMG
OBI+M2ePVvV6eNVvXXnutfD6f3n3n7bd1www3qqb7bd1www3q06ePJk+erLFjx769a9/rYsvvliTJk1SQ0ODR0oapVOnT
ql79+764x//qMsvv1//qMsvv1z19fVyu906ePBgyNcBAAAQaue91JufN6yjJEj2r2rZtm7Kysn7KysyTgwAGNHz9el112W
ahmPMeMGTN06NahpaWl6Uc/+tE5v+vdu7fGjx8ElGjx8VLcqyEcbjEAAALbvLQ9eKTt4ozFQHLIFQeEaKaOwnQhpcrL
ahmPMeMGTN06NAhpaWl6Uc/+tE5v+vdu7fGjx8ozy8vy7L01FNPqb6+Xv369dw4d9MQTT6ikpESlpaV69NFH9b
9PR0FRQUNP8ZOgAAgEh33h2/pKQkrVy5o888ojefffdUM30tbq7Vr10qSA+uvVn5+v9W6dOnq7q9SpSAo
GA+vbtK5fLpUGDBp3zucD27du3eC632628du3eC632628vDxJUkJ8QvCHBwAACLLzht/06tlCV6dNDNUeraGhoaP757Nmzz
Y/Pnj2rmJgRYdHq6amlRmpaV96Lq/XK6/XK6/XK6/XK6/NlMzWeUGhoaP757Nmzz
+vbt29rjQYAANCmhV34Jux5UqWlpQ0EAcBa28nKyh2Zy2y9e9r3+5FL7FHrfg9K0aPlrZ3Q4XEd0c0dcXJ46duyoTp06OnT
R8nQvwFXmcHgAAzHW+r3MJux0/AAAAfD0X9D1+xjsmdjAAAEDYY8cPAADAEIQfAAAAAIQg/AAAAAIQg/AAAAAIQg/AAAAAIQg/A
AAAAwR4/QAYaGbJI/TQwAh5nF6AABAa2PHDwAAwBCEHwAAAgCEIPwAAAEOEdfh16tRJ06ZN+1rHqpS
kaOTIka08EQAANsV1uHXuXnPfDAA1/r2NTUVI0aNaqVJwJ0a0NaqVJwIAAGi7wjr8Gs7Zs+eQUG
BysrKVFFRIY/1/HI0kaO3asNm3a1KxGqXtFlbjcHcdOHPGi6tyt3rcoTAI77mJSTpaLSjQ1cgtTVLa6sc2GA67NG66Dt9Q4fmPGB4uCVwQpUbtvpfmmF9pwkLm4m4R
BysrKVFFRIY/1/HI0kaO3asNm3a1KxGqXtFlbjcHcdOHPGi6tyt3rcoTAI77mJSTpaLSjQ1cgtTVLa6sc2GA67NG66Dt9Q4fmPGB4uCVwQpUbtvpfmmF9pwkLm4m4R
BysrKVFFRIY/1/HI0kaO3asNm3a1KxGqXtFlbjcHcdOHPGi6tyt3rcoTAI77mJSTpaLSjQ1cgtTVLa6sc2GA67NG66Dt9Q4fmPGB4uCVwQpUbtvpfmmF9pwkLm4m4R
BysrKVFFRIY/1/HI0kaO3asNm3a1KxGqXtFlbjcHcdOHPGi6tyt3rcoTAI77mJSTpaLSjQ1cgtTVLa6sc2GA67NG66Dt9Q4fmPGB4uCVwQpUbtvpfmmF9pwkLm4m4R
BysrKVFFRIY/1/HI0kaO3asNm3a1KxGqXtFlbjcHcdOHPGi6tyt3rcoTAI77mJSTpaLSjQ1cgtTVLa6sc2GA67NG66Dt9Q4fmPGB4uCVwQpUbtvpfmmF9pwkLm4m4R
BysrKVFFRIY/1/HI0kaO3asNm3a1KxGqXtFlbjcHcdOHPGi6tyt3rcoTAI77mJSTpaLSjQ1cgtTVLa6sc2GA67NG66Dt9Q4fmPGB4uCVwQpUbtvpfmmF9pwkLm4m4R

99brnnnuUkZGhgoICPfHEE/rJT36izZs368Ybb9TQoUM1d+5cxcfH6/7779fvf/97paWlKT09Xf/4xz90222
36dixY0pNTVWfPn30t7/9TZL0hz/8QQMGDFCfPn0UFxenrKwsSVJRUZG+//3vKy0tTY2Njc0zTZ06VbW1
tRowYIAsy5Lb7VZSUtLnZne73bJtW7ZtKyE+lZj/TAAAACER1PCrqqrSvn371NTUpMrKSr3++uuSpEAgoKS
kJI0YMUIzZsyQz+fT1q1bFRsbq6uvvlo7duzQE088occee0yJiYn66KOPFAgEdOutt2r27NnKyMjQhx9+KEk
aOnSodu7cKb/fr2HDhqlXr17q1KmTOnToJ07d0qSli1b1jzTiBEjdO+998rn82nXrl3q0qWLevTo8bnzZvV6
vLMuSZVmqrq8O5j8TAABASAT1C5wbGhqafz579mzzDkOt9B6zNNTkgjNsTjSimxWGLkDZ4rMLsPlVjtw5NBq
3T77bfrr3/9q77//e9ry5Yt6tevn0aNGqWf//znev311qxXZio2NPe9MUVFRys/P14YN
G1p/wQAAAG2Yozd3rF+/Xvn5+c2PU1NTJUnXXHONDh8+rHHz5mn16tXq27evunbtqvr6ei1dulRz585V
v379miOvurpaLpdLEyZMkCTV1taqraq5OAwYMkCTdddd57zntGnTFBPzWfP26NFD8fHxIVkAAAkxz9
k20/+9nP9Lvf/U5+v1/R0dGqqqrS6NGjNXHiRE2aNEmffPKJ3n33Xf3iF7+QZVmaO3euzp49euzp49q08++UTTpk
1TbW2tvF6v9u3bp3fffVe2bTefe+rUqfJ6vTp79qy2bdum2tpaSdKiRYuUlJSkPXv2KCoqSu+//77Gjh3r1D8
BAAABAyERJanJ6iGBwuVw6c+aMJOnxxx9X165d9fD3+tc9X165d9fDDD3+tc9lHbVleqzXHA74+j9MDAADaMtu2ZVktd4u
jO37BdPvtt+vHP/6xYmJidOTIEU2ZMsXpkQKAAABwVsTt+rel85QwAANCWnK9b+MsdAAAAhiD8AAAAD
EH4AQAAGILwAwAAMAThBwAAYAjCDwAAwBCEHwAAgCEIPwAAAEMQfgAAAIYg/AAAAAxB+AEAABi
C8AMAADAE4QcAAGCIGKcHCAvdJHmcHgLG8jg9AAAgUrDjBwAAYAjCDwAAwBCEHwAAgCEIPwAAAEMQfgAAAIYg/AAAAAxB+AEAABi
C8AMAADAE4QcAAGCIiAi/Rx55RIFAQIFAQIFAQA899JASExO1f/9RYWat++fVq/fr1iY2Mld27d9e6e6deu0/duFRcXq2fPng5PDwAAEBphH379
9+FRYWat++fVq/fr1iY2MlSd27d9e6e6deu0/duFRcXq2fPng5PDwAAEBphH379+vVTbm6ubrzxRt10001y
u9269NJL1aNHD/3xj39U7969VVNNTo+zsbEl9YVYWGh8vP6xhqKCs2fPrt10001y
XV19dLklatWqXMzExVVVWpoqJCklReXq6CS5XC4NGjRIK1eububD6+ffv2LZ7X7XYrLy9p9PkpkQQnxDkVQA
AAARf2IffF2loaGj+ubGDxGxUXFxcYqOjjOOjQzUs3S0tK+9Hiv1yuv1ytJso/aQ/AQ2rqUUm3AuYUg4
xcfHa9y4cSopKWnnxtXV1daqqqtKECROan+vbvt2+oRgUAAHBU2Iefz+fTc889Yp2zKeUe2SqVOnvv
D1OTk5mjp1qvvbu3avvKykqNGTMmhNMCAAAA4J0pSk9NDtHX2UVuW13J6DJjI4/QAAIBwY9u2LKvBM
HT8AAABcmIi9uaNVHRM7LwAAIOyx4wcAAGILwwAAAMAThBwAAYAjCDwAAwBCEHwAAgCEIPwAAAEMQfgAAAIYg/AAAAAxB+AEAABi
C8AMAADAE4QcAAGCIiAi/Rx55RIFAQIFAQIFAQA899JASExO1f/9RYWat++fVq/fr1iY2Mld27d9e6e6deu0/duFRcXq2fPng5PDwAAEBphH379
AMQfgBAAAYgvADAAAwBOEHAABgCMIPAAAAEMQfgAAAIYg/AAAAAxB+AEAABi
E8Tg8AADARO34AAACGIPwAAAAMQfgBAAAYos2GX6dOnTRt2rSvdExRUGys7ODNBEAAEB4a7Ph17l
zZz3wwAPn3wQHz3vwANOjwEAABAx2mz4EZZ49W9W9dee618Pp/p/mzJmjOXPXmKBAIYO/3a+LEic2vzmzEAAEAka7PhN2PGDB08eFBer1cPPPCAxx57
TDNk2r6KZhUdT5/Dnjs2ZbImMuvE5/iJw9N+a666NxG6O20ETNh6jNuH7dQ6qC8qrJQr4oc22PCXCT4m1Ek8vKlH6FyBBHU0d6VuGUbDiYjzNmhgaZ2aiEBq+y/cHWEPfsH1ulVn8iD+BsVl5LS1xKHAnYkh8zsjX3kVe3sbRuaLsP2bzZWoMbH+vV1ALuHx9kKTgDnuKiBaJ4Ls8kzmFxh7ZxcBqXq16T4s
0YcWcW2ZZmy5kLjqDBa4bWMhQnUyYgeZjPOnT/ANQzDlS4VHwvH1SbkX1tY1k3NcA2Ei2+uxKgZFVFgUt8WBptr9gFWGWM1kwVVuhVVZnbMmLrZ2p4gEKcWjtK5zeRfYQBN8hYrUd2xp6yYwqoGjLQqQ1D9U8GPVphgbnKe8
KJbLiRjwHFf9vFk2eStTd+gE9p8UnBvEcC4nF1VxlpmB48vwP9Us1OhKkRA4u3rqRyLaDgMPbRDp5uy+6L4jpSXBH9VSbaXWZyv5ccl+WxGSGjIS2pu9TaDYpdIhW8QKbFKHHXrcrAL7Ss+fOLgxwXN2A7fpcG5OkHqMhnd8QVhXprJaPbOVdPlhvHkWF2MuBqTtuiwcBAOD3GBYULOUGNJuRpXS+vOaABy2nTA5a4sMhxXlSyY8I8i/AMbQRnW4cPbkbRWq6gsYd3l3A1iHKunY5tKqwvGdMJ06zmWwqrx0B4nsfdP8Yn/Nt7G1tE/v0jI5jKyFIYdtm9nC+rwkj1H28AQBoYA4NkCsCbfMWNpx4MrqMAEYuZ+EfF9/mPVmB+q2i61xSqWAAdPUnn5Pgs42NCvYrKW94OmqnW7zs+BbVjh/9tvwtsmeGuVSYp9zdUOnx9CzVsYyBy7mwFCXUXlVJ3m8d6s3xg9qzj4AmVcnQ1kR2Zgnwm7yHMYF66nHLz/ifwU1ecjH4TNj3YBU4vVTH1W6GJ0Dx2Ntvqg5smZmnspZbQn6l5B57Kpu4Kmtg9z1FGTNn8jxbybxIR+PL6hEKfOWZe5nyQ+LuS9wZ6hT0e1ciY1pO8sGOfZyMM7UnFT4Qh6QeBK0+a9KbqoRRkrK5CJS3IUGF1xQ3AZthn1Do1B2eH6DJvk5gYddGTfXaeYWRN4Y6KGCS4gkU6nTxFQCyF2DGvUmW6fPa3gnZZ5Y4zDSv+XYMMBY+9X11I0mfrYrD5qI8gSXjsgeABaTlz2v5/wd6KXFsxRgNxN0KL1X6N0ZeC3B7

uIHAABgCIofAACAISh+AAAAhqD4AQAAGILiBwAAYAiKHwAAgCHCnA4QENpJSnc6BAJautMBAABgxQ8
AAMAYFD8AAABDUPwAAAAMEZTFLyIiQikpKXc8JjY2VsOHD6+nRAAAAM4LyuLXokUL/eQnP7njMT179
tSIESPqKREAAIDzgrL4ZWRkqHPnzsrPz9fhw4e1efNmSVKbNm106NAhdezYUS+88IKSkpKUn5+v8ePHO
5wYAADA94Ly61xmzZqlHj16KC4uTpL097//XdOnT9ewYcM0b948nThxQnPnzlXv3r01Y8aM214jOTlZU6
dOlSRFhkfWW3YAAABfCcri979mzJih4uJi7dy5U8uWLburc7KyspSVlSVJsk5ZvowHAABQL4LyUe//6tChg
2pqatS6dWuFhIQ4HQcAAMARQVn8qqqq1KxKmxZM0mSy+XSX//6Vz355JM6cOCCAnnvuuTrHHAAAAmCAoi
9/58+f1wQcfyOPx6ObNm3r//ff1wQcf6LnnntPTTz+trl27Kjc3V927d+fDHQQAAwBhB+x6/iRMn1tl3+fJlde
vWrXa7T58+9RkJAADAUUG54gcAAIC6gnbFz6vKJKU7HHQIAAOD+sOIHAABgCIofAACAISh+AAAAhqD4AQAAGILiBwAAYAiKHwAAgCEofgAAAIag+AEAABiC4gcAAGAIih8AAIAhKH4AAACGoPgBAAAYIszpAAGhnaR0p0MgIKU7HHQAAgP/Hih8AAIAhKH4AAACGoPgBAAAYwq+LX1RUDwej9MxAxAA
XP5fLpddee03FxcV655131Lhx
PjwYf3qV7+qvebEiRO1a9cu5efna9Gi
WppiY2Pl8Xg0b9682uOvX7X78ut9utRYs
UmTpxY5/cmJyfLsixZlqXmzp07FTzTUlJvl++32Kh
XrVsnSfJ4PNq3b59Onz4tSTp69Kg6duXuyaUhf2KSlLlJUgsCfOqr
Kgnfl/8rl27Vvvv6urq2se6XK6+/Rjt6uXmTRXMs0wlFRkcqZJ2nuVE5
aR0p0MgIKU7HQAAgP/Hih8AAIAhKH4AAACGoPgBAAAYgu
IHAABgiKAofuHh4dqwYnWhI6+9Wdt
PfLIIw6nBwAAYIsS
1ZcoUvfbaa+x2bdaa5o2bZqOHDmiPn36aHmDQj
OTo6WL1+uDRs26ZM+ePXr44YcZ
+oFAADA/aP4+9
1w/qDtkn+F
sAbrMmqd+wHTyP1ByDCXx98iQPGdPsBAqdiOB
QcEiXSnAwAAYxL1Je1lWjiuW/1Apc0wAATGdZ1oQ
DQAAUK+CYsUvJiZGLi
DtR+3HtYN4PQ5kWRoN5i4yqJ+BYp5kyhZxm+FRcmaJFcwsC3a0Poqa11yTp4c2rebEC7O
CQMXt6xiHTUl7IxnQO
+UK/cSQpTW9Pg8jnCDM8AAWbV9W4y5nbmfKNIwW3/FKlAsBVCFJS4tRm
WqvbJHXuM9mTpoO9xgvh3N2Zp3YjrXYPVNhyDz2UKvfqnX8m+8h5aXe3fiOpw5TvGNPOW/ymE
y8a7vpvRrz8x/Op5nTT48NbdHBD6L7vLKOEiVrUyvu3a71RtnCP+UOTCZ
YZnBIt6NC92yr+KHDlu
mAgJ1vB
qEP6+nLs0J1fgurBKQ5fMzm+/l5e7+PmVIEMB2JyMTcG
9dFmnmJ/ZUzT9A
jV/4TuPdhwbCDE
2A8G8pDZOyF9wWTYh3K4rWeVtHBcZuQnUUOhmqIVcBu99AScOrkNPOk3i3V5cgg3i7HyZ6DyMPaUuMD
OqSkfrTHalbj
5cAQG/0N0b6w1FTHj8QnM
nOHBx+4wjc2DH2GC3WnqN7/fv7+HSXyLsK9gnjyA
PLmn2/0hYb1WVJ3kAtP1eU9zl
/tU6xyLFEBvR4CCNXFnfULFf3xRU+Yt2s+pnhJwUjvgnB8Q2a2Myk0gfmHQwt9jxlfqzEtl9e6dBcMRCwgQdyEAhHEavgW1XhDuYKkrUj7
5hR6uM8mfeDcHHhTeERJR0ZVZXkz0YsjgNY2PTkpHRUUMVQoCkAXOQ/P5Ozik00Wq4kggSsv8tSYPwvL0W7D7TA
uIHAABgCIofAACAISh+AAAAhqD4AQAAGILiBwAAYAiKHwAAgCEofgAAAIag+AEAABiC4gcAAGAIih
8AAIAhKH4AAACGoPgBAAAYIszpAAGhnaR0p0MgIKQ7HQAAgM/Hih8AAIAhKH4AAACGoPgBAAAYgu
AAA2TA4mzEFZrBGQEaDugsxVQWCjgFE3eK/ERlYwxBTDbbrb3+GjYVFLWBtPCVxn0E9wxO0ebkY
DmjyE2zbAs/Wy1/E3YscsNi4GW0hZ+brvz6WnuhQw+kzaOZ1CAAUGDtAUmZBYdr4qKL1iD9XFtxGiwg/JtR6XJ
PgBGwZmJMvWEVEf7YNRmTVO/KSJrMkatt8BbJ2uaUJZN5UVBDUbrx1upCfG77Qg0OE0UUZ3LuL0yREnk8Yx8mI5
iPnbrYgVRz6Np08Cwu/K2hwCO0iQxd4OzTWvTXw+x/v8P2f3U8hhDiAQCJ2/Q5jqo3JRUmBnRMlK4rV7HoW73Eq
ZNt/fW1c4Sp8MzuxqLFRfAl5pCQk3pymhbA/hw/3MkVvmnHrU33xkTw7pSWcpK2/OA8mXZ7U4s/VtBYJkqeIp7rUG
iGQ8g47Yg1L0eSC1QCW8hM3UcNVL0cKhjkYJcAWpXcQqXk2aZYsfBOxGAiW91cQcO8VQ5fGQaeKQWv7p9/LIt
gC6+X92kR4K+zzB0nNNk8kx6Dv7PU00/k4+jVPuTDPwhm18lEJ9v0sNMsLTLKp3d+4/MMIsC6dY0pANHu9iwC
UbMANutQGqafCu93d0ZoUbu7M8jsE6SEbxNCX1eZPMq0YiCghTPMqEk4FzZO0VBoHXVfMU/0Mdhge
r6R6dmdHAJtDNANEX0w6F4d6DS5Cx35+l/yQdQnP7jzLTvl7vJwz6vS6X9+zHxF/4F7Fu+fr/JZUGDJS0
eDJZG2HDqz6d+0jD+1Jm1cMNNXqKY9L4uGpZ8WKDG6qUKqUb0m0lUUEQ88GZl7AVxRY/A5ZaG8dsxRx6i
0ibJgG2M9HMnxFOYdCjPMZEoxPPzEqeOtgtW83bv4J31WzpbrHJw4PQJYx3+R4SAV16GTXq1Aolks2lPt
ERKV6ODR7PMqJ0A0PRYg0Wc2CmE8x6JBmm5OmDMbz92iBsaCcUPhyRh1ok3LQB2IbJ2bhmAhuMEq/4kHU
/CKHcp8fP7I89ZWu+xY1HEeIsoOJ8y09LoiiBUgAaePLCqyhFaKGM5QKZ19kSHc+vZJaxdYCr0zCOW6W
7RcpgQlt6+GaHMOuJ4/vCGp5nZxzW9JQHyyN3SPIZ3JQt6x0rSbrbKkzYXcQfaB8WGPD/XvQ9AKEj8W
0XJnRhR/2LKbhhO7xtlwaDuCu8oAK/H9MvZVk6PwzObbx1SdwvjX7ZmtxFmR+iK/QbGZNYKXg7CWBo
b7JXXdZfYQuIMDgPTAkPSQ8DjAyWUcRiH6kAAuQC8N7SNuNv1T4SOPpr5d49RpGMYnlqhwA1UbB
3GxBIB/xEwGRPf+Ps+NgJG4xsvKwfDYcHMm2mU/LQO3kgOCpVu6z2V4WE9HX6E4dJbfqAB6GNRs
kBdTWbwTJTBVxUpW3PWt4hDQmLlHSGFlNOQS1L0nkfKF4KmUJV1udqBuoVJoPiFvX4TrFf83oiBmYf
8pkb9vo5Q6jWzQ3U3BVbsHVwD5AV7jGvrNaSZcSHu39wKO8g1SE6A+9KPSsYG2CJeFBH2ydaoOyyg
ss0m9ErG4P8gZpTcNdMlZ4jkxYVpI94Va7hmw5sUQtRkkRhKf4ZjnpH71H9cMRhFE9DqBjRkTgVkI
m6PRwPbUJXbp8TyE/Lq1Zlq6Dj4ke02jYRSQkVrb9cJuHY9rlLDgjiQgBjmm5axSTgkqCSLEzUgz
ZR5A7n4TbTL5k0h56yGK4dNqm8d8YpFRZHy0LK2SL9/u3rd1LkQ5JDYaORHBMy/ifxRcsbykcYMwI
5qGD3hD+n6Uh9vUdC0LF0Vmdk7wzIy1fEqxx8N+qb5YEeFhsRxrLUrBAf/jl2v7tWdXezDvxUqlq/
+7hIr7X6KmxzA3tdUmVg7/Sz5tpX1rQNewVLK4GXvjhpDPc6sSTIjRcGy5ADBZAdKr0MM85JzKL4gyL
8fp8NFpoL/4fzkDCmGzRzQTQDP/oG3ugpRKwsOlT0hyO6PiKtT3rt1qYAp6L2eQcoBKBIWGO3yG9
8EBqX75hK/eQz3t0uHvkMtYNCEiJF7TC0ZCMJM3nj3nIEeYkjAtvmwtPJNnE3t1xhZmymAE+9b8F
YNx/Q59a9TKfD0RC/jAE0sDBQX6yjOyt0xX1BDpAzNPS0WXJNhOVVIhV5sjRP8nI3rGmh5gVx2C
OLXfPtw0WNNmM1J/xDf9b5xEkvbhtsSo7U5KmiYsVGW9VVA/eeHKu/0H+tVTXyIXSaAG9fZvkcI
4Q1BXtdhPxNrmZDZXGuSVHjVYC5bOeo90EFnJqzXW1FgtR9SmHjwYSXYiWnqzYSFLnaSTOGS/VvY
j2k5nfbzmuUl/N3YIJpXmJzvMhPKgRg7xHrXv+f/gXkq3MW4K2gRhl18xX7G2ZEK8vZQ8TZgaZCc
nKM4RT6dZ2FQDbdq4mOepfxVOfITZXt4ObxFXJq8Rp+vR5jBOBZrzJjJ2xDRZKrG2SVmQLdMVpHx
evVw8/9E3dvxPmnQjdPeO2Nds9/cEw1kMYZw/gLIj3GfQ2Q7nqkp7XcJ/1nb+PdnC/ujEKl5zXLYF
BmQrw4RwzA/9O6X3DoGc/aJHoV1J66W9n3XyjLd94Lx5Mmxn/MV+ptC+6GVMLjnRxnX9ka3UE5EQs
EgxUeyEeskvrL1zNeKNXmPQ1s4awqdRuK4K0hVkB73MkFfFI8WfgtTBQtN11ll4QHq9ZX33b4M+m3W
zZZTQHsfDwfXqu6aLd3pvcnV2t1qodgtm3ZOdCAcXnmjpTmnvcvVU8W74Xa+xONv9ZDV0nLOSVxqxO
rQJlwN1MmSe2qXl+2zEDTRDb9L4HhkcPsZ0HJZX0OVV2TdzbrVlK1sY1gbEpZz7p3o8i8AOGgj0T9h
LSnHSV2HvtlJy8nMaF5qUbtJ5ILuVO8eNJkrBmbnkz8ZLP8DwKYm6ytAGbaa5nrb5oaTF7RJMXM/A
+tvaz4oNWi0Hh3mJtaeUNJqH+4p8ar/JzUfn7m/7M0KNxQa/mQjn5uqVDFTLXHvUHYEEWwJb7b+qeY
AAhEhKFGQbWvA+kLuO2GgbqAQnTdJAxmN6VEmsSEcl1YI5Mw6E+cDQvNj/RnIgQN8WJQJQE7zqFOxM
FxJSr5ta8PL8Nvfdv9Y3I6pHW/yJ0Jvou3qdJDjA7sCP9b/PPcsGYSOFhuFbg9WbT+u+eJk+K+a/rXh
A3WuvsEMmlAsTkOupH0XeH1CbnMqAnMwaiVw==

AAAYguIHAABgCIofAACAISh+AAAAhghzOkBAaCcp3ekQ8HvpTgcAAODOWPEDAAAwBMUPAADAEBQ/
AAAAQwRN8XvmmWfUpEkTp2MAAAD4raApfqmpqQoPD3c6BgAAgN8KyOIXHh6uDRs2qKCgQB6PR3
PnzlW7du2Um5urLVu2SJImTJigoqIieTweZWRk1J5bVVWll19+WcXFxdq8ebMiIyOdug0AAIB6FZDFb9iw
YSorK1PPnj0VExOjzMxMlZWVafDgwUpMTFTbtm314osvKjExUT179pTb7dbo0aMlSU2bNtWePXvUo0c
Pbdu2TfPmzbvt70hOTpZlWbIsS5HhlEMAABD4ArL4eTweDR06VBkZGUpISNClS5du+bnb7dbWrVtVUV
Gh6upqZWdna+DAgZKk6upqqLV++XJL0j3/8QwkJCbf9HVlZWXK73XK73aq4WuHbGwIAAKgHAfkFziUlJY
qPj9eIESM0f/58vffee1/6WrzEZAACA/wrIFb+2bdvbxdvq6tWrys7O1oIFCxQfH6+qqio1a9ZMkrR7924NGj
RIrVq1UmhoqJ588ZNkuRyuTRu3DhJ0ve//33t2LHDsfsAAACoTwG54hcTE6MFCxaopqZGN27cUEp
Kivr3769NmzaprKxKxMiYmJjJmjVrlnJzcxUSEqKNGzdq3bp1kqTLLy+rT58+mjNnjsrLy5WUlOTw3QAAANSPE
ElGPev87Mrg3bJOWXJnuX2UCEEj3ekAAABIlmXJ7b59bwnIR70AAAC4dwH5qPd+3OtqnySpKmACA
gMeKHwAAgCEofgAAAIag+AEAABiC4gcAAGAIih8AAIAhKH4AAACGoPgBAAAYguIHAABgCIofAACAISh+
AAAAhqD4AQAAGILiBwAAYIgwpwMEhHaS0p0OAb+V7nQAAADuDit+AAAAhqD4AQAAGILiBwAAYAiK
HwAAgCEofp8REhLidBBg1q6dKmKi4vVsBZMUfd9VfNHpyMBAAD4jPFf59RiDKbmKwQmUNdP4dHmz
btumV/cnKypk6d6dKmKDI90OquAAMDX+bwXPx2/NOQ+HFbYn1zbjYZwYVq6dEnZ8mT+1u+f+1W/+IYee+mkbx
HQEAAAAAMQFD8AAAAAwBMUPAADAEBQ/AAAAAMAQFD8AAAAAwBMUPAADAEBQ/AAAAAMAQFD8AAAAAwBMU
Py+wLe//nP+6Ptn0dfnbd7D8LSPWQ6dr/eNWBDI4yGmJOeJDoiuBz4TM/VJQLjk0GvTBmtTHoPB+zPlvx+fYn5o0Pnz
9y4n5jYqK0le+8pXP/bnXOwLMvxDME8mF/mN5AH88scB/pgfs2aXx71AgAAGILiBwAAAGILiBwAAGILiBwAAYAiK
Xv3r1ORwhqzK9vMb+++xfz6HnPsW8yvb/nT/Lhat/LPLhDgAAAAEPwqBcAAAAQFD8AAAAAwBMUPAADAEBQ/
hQJSUlSktLzcpzpOQFq8eLHnDkjj8djDj8dTu69du3YatmyZbFarYqOjtaoUaN8fj8dTu69ly5bpKycncR4ZjzZ
FDB23ZskX79u1TxXKfdnFVg0FjenLjZiRVbrlHqTGHvUdUwsyC0dLloMdqqyBbc0t6PFLVYOzOJVaEybtt0WGfcgBvFFhgzYA0q
WrZsmUpKSRz505505FRUU5mD5wD5whIaGau/evSrPi4wFkNBZVQtLeWNnLVPOnF8R2bfuF2B0l+cBObYFqSJSbPyYnnQG+
e3fv1/9+vXz+/vz+vXz/+l1/DtI/HWEhobaR44csJ0/Ojo0+0GDrYBQUFdru3RzPFWjj8ccft+Pi4myPx1O77778UXX7TTtO
tJsSXZaWpqdkZFhS7KHDx9u/+tf/7/7Il2X379rV37tzpeH5/H23atLHQDRQUGOQvfaEV/4KkpH8yKTVZL/KbJz5LQ
OABWSIdFhZm79y/7byHSzoyOkpZbrjOE7KTkpLsZZJZLbe5Fbeo5Mbry69l4v1pjOjtCkPl9/2Dqr6ZZ4DzHzk5LuU
Z2dn2+vXr7clMb9ePHeKWlpXarVq1u2cfrg/fG66+/bk+ZMsWdDdo0MCOiIjw9/l1ftL8dfTr18/etGlT7fas
WbPsWbNmOZ4rEEdNDUVNQtxe/gwYN2mzZtbOmT4nLw4EFbkr1o0S3WoQJtz2Oc/x3o9zq/sml
BaNKkiZ2Xl2f36dPPnv2rO1yuwp1teKTZs22f3696bM2l2S6Xyz579qzjuf/tG/f3t68ebM9ePDg2uLH/
v3K748frqndG8eXP76NGjdfb78/zcxfr0xkU3l5Uu3bt3cwNnqq1Tp8/+LUk6ffq0L

k5v19RUVGKi4vTrl27mGMvCg0NVX5+vsrLy/Xuu+/qo48+UmVlpaqrqyXdOoefnd/q6mpdvHhRrVq1cix
7IMjMzNTMmTNVU1MjSWrVqhXz60W2bSsnJ0d79uxRcnKyJF6DvSU6Olpnz57VkiVLtHfvXmVlZSk8PN
yv55fiB79g27bTEQLeAw88oFWrVik1NVVVVVV1fs4cf3k1NTWKi4tThw4d1KdPH3Xt2tXpSEFj5MiRKi8v
96vvOQs2CQkJ6tWrl4YPH67p06fr8ccfr3MMrw9fTlhYmOLj47Vw4ULFx8frypUrmjVrVp3j/Gl+KX53cOr
UKXXs2LF2u0OHDjp16pSDiYLHmTNn1KZNG0lSmzZtVF5eLok5/7VdlWwvG/vdJvVhR9QcqTT5Hea+3ZOVrS2ylVnD/W/ZyXJgYHe
icnNz1b9/f7Vo0UIul0vSrXP42fl1uVyKiIjQuXPnHMvs7wYMGKDvf0Ki0t1bJly5SYmKhXXnmF+fWisrIy
SdLZs2e1Zs0a9enTh9cHLzl58jqyp3bt3S5LeeustxcfH+/X8UvzuwLIsdenSRZ06dVKDBg00YcIErVu3z
ulYQWHdunWaNGmSJGnSpElau3Zt7f4f4f/vCHkqS+ffvq4sWLtcvl+HyLFy/WgQMH9Lvf/a52H3PsHZGk
qIiJAkNW7cWEOHDtWBAweUm5urcePGSao7v5+urcePGSao7v5/O+7hx47RlyxZngeI2bNnq2PHjoqOjtaECRO0ZcsW/e
AHP2B+vSQ8PFxmmzat/fe3/rgUtFRcX8/rgJWfOnNGJEyf09a9a9/XZI0ZMgQ7d+/+/3/E3R/rzGD58uH3o
0CH7yjEj9uzZsx3PE4xfTfGbxvOMvKyuzr16/bz4aaWqZazu4OGD7yYjyjb3+eefty
PnLkiF1UVGT36tXz+PFmGrt3rrt8YWGh8b5PB77+eefty
XZ0dHR9q5du+ySkJ7xYoVdsSJH7XtnUGV9q5du+zo6GjH7yFQxqBBg2o/MAX+0dHmdER
0fbBQUFdkFBgV1cXFz7/xivD94bsGxtmVZdmFhob1mzRq7RYsWzNq7RYsWfj2//Mk2AAAAQ/CoFwAAwBAUPwA
AAENQ/AAAAAxB8QMAADAExQ8AAMAQFD8AAABDUPwAAAAM8X9Kx5+pZsPXLQAAAABJRU5ErkJgg
g==\n"

        },
        "metadata": {}
      }
    ]
  },
  {
   "cell_type": "code",
   "source": [
    "from sklearn.preprocessing import LabelEncoder\n",
    "lb_enc = LabelEncoder()\n",
    "y = lb_enc.fit_transform(df[\"SpamHam\"])"
   ],
   "metadata": {
    "id": "wzRr_hm258UC"
   },
   "execution_count": 52,
   "outputs": []
  },
  {
   "cell_type": "code",

```
 "source": [
  "tokenizer = Tokenizer() #initializing the tokenizer\n",
  "tokenizer.fit_on_texts(X)# fitting on the sms data\n",
  "text_to_sequence = tokenizer.texts_to_sequences(X)"
 ],
 "metadata": {
  "id": "Achf5DUl5_0s"
 },
 "execution_count": 53,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "for i in range(5):\n",
  "    print(\"Text              : \",X[i] )\n",
  "    print(\"Numerical Sequence : \", text_to_sequence[i])"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "nbygskt26EDp",
  "outputId": "e798c220-4eb1-4e26-9645-38ea99f55303"
 },
 "execution_count": 54,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
```

```
        "Text           :  go jurong point crazy available bugis n great world la e buffet cine got amore wat\n",

        "Numerical Sequence :  [3, 3575, 255, 549, 522, 1028, 35, 56, 236, 845, 80, 2411, 1029, 9, 3576, 70]\n",

        "Text           :  ok lar joking wif oni\n",

        "Numerical Sequence :  [6, 220, 1173, 328, 1539]\n",

        "Text           :  free entry wkly comp win fa cup final tkts st may text fa receive entry questionstd txt ratetcs apply over\n",

        "Numerical Sequence :  [5, 342, 582, 724, 92, 1540, 846, 478, 1541, 147, 175, 17, 1540, 221, 342, 2412, 23, 2413, 263, 2414]\n",

        "Text           :  dun say early hor c already say\n",

        "Numerical Sequence :  [142, 48, 256, 2415, 69, 75, 48]\n",

        "Text           :  nah think go usf life around though\n",

        "Numerical Sequence :  [774, 33, 3, 725, 86, 121, 195]\n"

       ]

      }

     ]

    },

    {

     "cell_type": "code",

     "source": [

      "tokenizer.index_word"

     ],

     "metadata": {

      "colab": {

       "base_uri": "https://localhost:8080/"

      },

      "id": "KnoF_1oW6IcP",

      "outputId": "5ec6d556-a48a-4264-a109-bee36f418227"

     },

     "execution_count": 55,

     "outputs": [
```

```
{
 "output_type": "execute_result",
 "data": {
  "text/plain": [
   "{1: 'call',\n",
   " 2: 'get',\n",
   " 3: 'go',\n",
   " 4: 'know',\n",
   " 5: 'free',\n",
   " 6: 'ok',\n",
   " 7: 'ltgt',\n",
   " 8: 'day',\n",
   " 9: 'got',\n",
   " 10: 'want',\n",
   " 11: 'come',\n",
   " 12: 'like',\n",
   " 13: 'love',\n",
   " 14: 'going',\n",
   " 15: 'good',\n",
   " 16: 'time',\n",
   " 17: 'text',\n",
   " 18: 'send',\n",
   " 19: 'message',\n",
   " 20: 'need',\n",
   " 21: 'one',\n",
   " 22: 'today',\n",
   " 23: 'txt',\n",
   " 24: 'home',\n",
   " 25: 'lor',\n",
   " 26: 'see',\n",
   " 27: 'sorry',\n",
```

" 28: 'stop',\n",

" 29: 'r',\n",

" 30: 'still',\n",

" 31: 'back',\n",

" 32: 'mobile',\n",

" 33: 'think',\n",

" 34: 'reply',\n",

" 35: 'n',\n",

" 36: 'take',\n",

" 37: 'tell',\n",

" 38: 'phone',\n",

" 39: 'new',\n",

" 40: 'week',\n",

" 41: 'later',\n",

" 42: 'da',\n",

" 43: 'hi',\n",

" 44: 'please',\n",

" 45: 'cannot',\n",

" 46: 'make',\n",

" 47: 'tomorrow',\n",

" 48: 'say',\n",

" 49: 'night',\n",

" 50: 'claim',\n",

" 51: 'well',\n",

" 52: 'thing',\n",

" 53: 'much',\n",

" 54: 'dear',\n",

" 55: 'oh',\n",

" 56: 'great',\n",

" 57: 'hey',\n",

" 58: 'pls',\n",

" 59: 'would',\n",

" 60: 'give',\n",

" 61: 'number',\n",

" 62: 'way',\n",

" 63: 'happy',\n",

" 64: 'friend',\n",

" 65: 'work',\n",

" 66: 'hope',\n",

" 67: 'u',\n",

" 68: 'let',\n",

" 69: 'c',\n",

" 70: 'wat',\n",

" 71: 'prize',\n",

" 72: 'right',\n",

" 73: 'yes',\n",

" 74: 'min',\n",

" 75: 'already',\n",

" 76: 'ask',\n",

" 77: 'really',\n",

" 78: 'yeah',\n",

" 79: 'said',\n",

" 80: 'e',\n",

" 81: 'k',\n",

" 82: 'babe',\n",

" 83: 'co',\n",

" 84: 'p',\n",

" 85: 'amp',\n",

" 86: 'life',\n",

" 87: 'meet',\n",

" 88: 'thanks',\n",

" 89: 'morning',\n",

" 90: 'year',\n",

" 91: 'last',\n",

" 92: 'win',\n",

" 93: 'miss',\n",

" 94: 'service',\n",

" 95: 'cash',\n",

" 96: 'find',\n",

" 97: 'ill',\n",

" 98: 'feel',\n",

" 99: 'lol',\n",

" 100: 'anything',\n",

" 101: 'every',\n",

" 102: 'tone',\n",

" 103: 'nokia',\n",

" 104: 'sure',\n",

" 105: 'pick',\n",

" 106: 'also',\n",

" 107: 'soon',\n",

" 108: 'b',\n",

" 109: 'contact',\n",

" 110: 'keep',\n",

" 111: 'something',\n",

" 112: 'care',\n",

" 113: 'sent',\n",

" 114: 'urgent',\n",

" 115: 'could',\n",

" 116: 'buy',\n",

" 117: 'place',\n",

" 118: 'next',\n",

" 119: 'first',\n",

" 120: 'guy',\n",

" 121: 'around',\n",

" 122: 'even',\n",

" 123: 'tonight',\n",

" 124: 'show',\n",

" 125: 'went',\n",

" 126: 'customer',\n",

" 127: 'help',\n",

" 128: 'someone',\n",

" 129: 'wish',\n",

" 130: 'per',\n",

" 131: 'wait',\n",

" 132: 'many',\n",

" 133: 'nice',\n",

" 134: 'money',\n",

" 135: 'chat',\n",

" 136: 'ya',\n",

" 137: 'sleep',\n",

" 138: 'wan',\n",

" 139: 'word',\n",

" 140: 'always',\n",

" 141: 'late',\n",

" 142: 'dun',\n",

" 143: 'people',\n",

" 144: 'leave',\n",

" 145: 'name',\n",

" 146: 'end',\n",

" 147: 'st',\n",

" 148: 'v',\n",

" 149: 'waiting',\n",

" 150: 'gud',\n",

" 151: 'told',\n",

" 152: 'hour',\n",

" 153: 'tc',\n",

" 154: 'x',\n",

" 155: 'lot',\n",

" 156: 'minute',\n",

" 157: 'getting',\n",

" 158: 'girl',\n",

" 159: 'guaranteed',\n",

" 160: 'yet',\n",

" 161: 'thk',\n",

" 162: 'coming',\n",

" 163: 'box',\n",

" 164: 'month',\n",

" 165: 'done',\n",

" 166: 'hello',\n",

" 167: 'smile',\n",

" 168: 'try',\n",

" 169: 'thought',\n",

" 170: 'person',\n",

" 171: 'haha',\n",

" 172: 'talk',\n",

" 173: 'class',\n",

" 174: 'best',\n",

" 175: 'may',\n",

" 176: 'live',\n",

" 177: 'mean',\n",

" 178: 'offer',\n",

" 179: 'th',\n",

" 180: 'fine',\n",

" 181: 'finish',\n",

" 182: 'lunch',\n",

" 183: 'man',\n",

" 184: 'job',\n",

" 185: 'line',\n",

" 186: 'stuff',\n",

" 187: 'use',\n",

" 188: 'car',\n",

" 189: 'sm',\n",

" 190: 'bit',\n",

" 191: 'draw',\n",

" 192: 'ppm',\n",

" 193: 'holiday',\n",

" 194: 'heart',\n",

" 195: 'though',\n",

" 196: 'yup',\n",

" 197: 'trying',\n",

" 198: 'never',\n",

" 199: 'special',\n",

" 200: 'meeting',\n",

" 201: 'plan',\n",

" 202: 'long',\n",

" 203: 'better',\n",

" 204: 'account',\n",

" 205: 'dat',\n",

" 206: 'start',\n",

" 207: 'chance',\n",

" 208: 'cool',\n",

" 209: 'ready',\n",

" 210: 'problem',\n",

" 211: 'god',\n",

" 212: 'mind',\n",

" 213: 'latest',\n",

" 214: 'weekend',\n",

" 215: 'birthday',\n",

" 216: 'hr',\n",

" 217: 'room',\n",

" 218: 'guess',\n",

" 219: 'enjoy',\n",

" 220: 'lar',\n",

" 221: 'receive',\n",

" 222: 'cost',\n",

" 223: 'eat',\n",

" 224: 'look',\n",

" 225: 'half',\n",

" 226: 'play',\n",

" 227: 'check',\n",

" 228: 'awarded',\n",

" 229: 'nothing',\n",

" 230: 'game',\n",

" 231: 'real',\n",

" 232: 'yo',\n",

" 233: 'another',\n",

" 234: 'big',\n",

" 235: 'voucher',\n",

" 236: 'world',\n",

" 237: 'camera',\n",

" 238: 'landline',\n",

" 239: 'house',\n",

" 240: 'boy',\n",

" 241: 'shit',\n",

" 242: 'probably',\n",

" 243: 'dinner',\n",

" 244: 'sweet',\n",

" 245: 'sir',\n",

" 246: 'liao',\n",

" 247: 'jus',\n",

" 248: 'kiss',\n",

" 249: 'ever',\n",

" 250: 'watching',\n",

" 251: 'pm',\n",

" 252: 'watch',\n",

" 253: 'might',\n",

" 254: 'quite',\n",

" 255: 'point',\n",

" 256: 'early',\n",

" 257: 'xxx',\n",

" 258: 'speak',\n",

" 259: 'aight',\n",

" 260: 'called',\n",

" 261: 'po',\n",

" 262: 'pic',\n",

" 263: 'apply',\n",

" 264: 'fun',\n",

" 265: 'bus',\n",

" 266: 'nd',\n",

" 267: 'part',\n",

" 268: 'hear',\n",

" 269: 'pa',\n",

" 270: 'bed',\n",

" 271: 'rate',\n",

" 272: 'thanx',\n",

" 273: 'video',\n",

" 274: 'baby',\n",

" 275: 'two',\n",

" 276: 'actually',\n",

" 277: 'den',\n",

" 278: 'princess',\n",

" 279: 'network',\n",

" 280: 'remember',\n",

" 281: 'pay',\n",

" 282: 'maybe',\n",

" 283: 'forgot',\n",

" 284: 'worry',\n",

" 285: 'question',\n",

" 286: 'easy',\n",

" 287: 'nite',\n",

" 288: 'reach',\n",

" 289: 'shopping',\n",

" 290: 'orange',\n",

" 291: 'bad',\n",

" 292: 'code',\n",

" 293: 'pobox',\n",

" 294: 'left',\n",

" 295: 'dis',\n",

" 296: 'ah',\n",

" 297: 'hurt',\n",

" 298: 'award',\n",

" 299: 'little',\n",

" 300: 'sat',\n",

" 301: 'put',\n",

" 302: 'shall',\n",

" 303: 'fuck',\n",

" 304: 'dad',\n",

" 305: 'selected',\n",

" 306: 'enough',\n",

" 307: 'looking',\n",

" 308: 'school',\n",

" 309: 'yr',\n",

" 310: 'working',\n",

" 311: 'evening',\n",

" 312: 'leh',\n",

" 313: 'g',\n",

" 314: 'face',\n",

" 315: 'everything',\n",

" 316: 'gift',\n",

" 317: 'made',\n",

" 318: 'mate',\n",

" 319: 'pound',\n",

" 320: 'thank',\n",

" 321: 'afternoon',\n",

" 322: 'wk',\n",

" 323: 'without',\n",

" 324: 'tv',\n",

" 325: 'xmas',\n",

" 326: 'asked',\n",

" 327: 'movie',\n",

" 328: 'wif',\n",

" 329: 'ringtone',\n",

" 330: 'since',\n",

" 331: 'came',\n",

" 332: 'anyway',\n",

" 333: 'must',\n",

" 334: 'join',\n",

" 335: 'sexy',\n",

" 336: 'dream',\n",

" 337: 'missing',\n",

" 338: 'sound',\n",

" 339: 'collect',\n",

" 340: 'detail',\n",

" 341: 'town',\n",

" 342: 'entry',\n",

" 343: 'uk',\n",

" 344: 'lesson',\n",

" 345: 'wanted',\n",

" 346: 'run',\n",

" 347: 'okay',\n",

" 348: 'price',\n",

" 349: 'pmin',\n",

" 350: 'answer',\n",

" 351: 'mail',\n",

" 352: 'able',\n",

" 353: 'hav',\n",

" 354: 'important',\n",

" 355: 'wake',\n",

" 356: 'bring',\n",

" 357: 'collection',\n",

" 358: 'mob',\n",

" 359: 'test',\n",

" 360: 'juz',\n",

" 361: 'til',\n",

" 362: 'stay',\n",

" 363: 'true',\n",

" 364: 'charge',\n",

" 365: 'plz',\n",

" 366: 'book',\n",

" 367: 'office',\n",

" 368: 'change',\n",

" 369: 'kind',\n",

" 370: 'away',\n",

" 371: 'plus',\n",

" 372: 'wife',\n",

" 373: 'update',\n",

" 374: 'till',\n",

" 375: 'pain',\n",

" 376: 'hair',\n",

" 377: 'wen',\n",

" 378: 'else',\n",

" 379: 'age',\n",

" 380: 'weekly',\n",

" 381: 'wot',\n",

" 382: 'dude',\n",

" 383: 'attempt',\n",

" 384: 'de',\n",

" 385: 'valid',\n",

" 386: 'colour',\n",

" 387: 'credit',\n",

" 388: 'alright',\n",

" 389: 'saw',\n",

" 390: 'yesterday',\n",

" 391: 'double',\n",

" 392: 'trip',\n",

" 393: 'making',\n",

" 394: 'food',\n",

" 395: 'top',\n",

" 396: 'bt',\n",

" 397: 'haf',\n",

" 398: 'si',\n",

" 399: 'ltdecimalgt',\n",

" 400: 'oso',\n",

" 401: 'missed',\n",

" 402: 'music',\n",

" 403: 'bored',\n",

" 404: 'lei',\n",

" 405: 'date',\n",

" 406: 'ard',\n",

" 407: 'tried',\n",

" 408: 'delivery',\n",

" 409: 'driving',\n",

" 410: 'hot',\n",

" 411: 'friendship',\n",

" 412: 'drink',\n",

" 413: 'either',\n",

" 414: 'sch',\n",

" 415: 'online',\n",

" 416: 'brother',\n",

" 417: 'mom',\n",

" 418: 'second',\n",

" 419: 'bonus',\n",

" 420: 'address',\n",

" 421: 'player',\n",

" 422: 'order',\n",

" 423: 'smoke',\n",

" 424: 'story',\n",

" 425: 'id',\n",

" 426: 'wid',\n",

" 427: 'club',\n",

" 428: 'full',\n",

" 429: 'head',\n",

" 430: 'tot',\n",

" 431: 'sae',\n",

" 432: 'lose',\n",

" 433: 'family',\n",

" 434: 'together',\n",

" 435: 'feeling',\n",

" 436: 'goin',\n",

" 437: 'sad',\n",

" 438: 'set',\n",

" 439: 'close',\n",

" 440: 'smiling',\n",

" 441: 'old',\n",

" 442: 'believe',\n",

" 443: 'noe',\n",

" 444: 'happen',\n",

" 445: 'walk',\n",

" 446: 'calling',\n",

" 447: 'post',\n",

" 448: 'huh',\n",

" 449: 'beautiful',\n",

" 450: 'tscs',\n",

" 451: 'saying',\n",

" 452: 'drive',\n",

" 453: 'await',\n",

" 454: 'aft',\n",

" 455: 'took',\n",

" 456: 'finished',\n",

" 457: 'busy',\n",

" 458: 'private',\n",

" 459: 'match',\n",

" 460: 'leaving',\n",

" 461: 'everyone',\n",

" 462: 'hand',\n",

" 463: 'gr',\n",

" 464: 'awesome',\n",

" 465: 'shop',\n",

" 466: 'coz',\n",

" 467: 'ring',\n",

" 468: 'okie',\n",

" 469: 'mths',\n",

" 470: 'eve',\n",

" 471: 'land',\n",

" 472: 'thinking',\n",

" 473: 'mum',\n",

" 474: 'email',\n",

" 475: 'w',\n",

" 476: 'rite',\n",

" 477: 'drop',\n",

" 478: 'final',\n",

" 479: 'caller',\n",

" 480: 'forget',\n",

" 481: 'congrats',\n",

" 482: 'started',\n",

" 483: 'statement',\n",

" 484: 'taking',\n",

" 485: 'content',\n",

" 486: 'break',\n",

" 487: 'company',\n",

" 488: 'wil',\n",

" 489: 'angry',\n",

" 490: 'unsubscribe',\n",

" 491: 'card',\n",

" 492: 'sister',\n",

" 493: 'valentine',\n",

" 494: 'reason',\n",

" 495: 'poly',\n",

" 496: 'simple',\n",

" 497: 'neva',\n",

" 498: 'pub',\n",

" 499: 'possible',\n",

" 500: 'treat',\n",

" 501: 'news',\n",

" 502: 'tomo',\n",

" 503: 'lucky',\n",

" 504: 'search',\n",

" 505: 'expires',\n",

" 506: 'open',\n",

" 507: 'fancy',\n",

" 508: 'bank',\n",

" 509: 'worth',\n",

" 510: 'touch',\n",

" 511: 'found',\n",

" 512: 'sleeping',\n",

" 513: 'choose',\n",

" 514: 'mine',\n",

" 515: 'whatever',\n",

" 516: 'sun',\n",

" 517: 'knw',\n",

" 518: 'anyone',\n",

" 519: 'parent',\n",

" 520: 'alone',\n",

" 521: 'auction',\n",

" 522: 'available',\n",

" 523: 'winner',\n",

" 524: 'ha',\n",

" 525: 'smth',\n",

" 526: 'saturday',\n",

" 527: 'decided',\n",

" 528: 'song',\n",

" 529: 'ticket',\n",

" 530: 'prob',\n",

" 531: 'happened',\n",

" 532: 'uncle',\n",

" 533: 'unredeemed',\n",

" 534: 'identifier',\n",

" 535: 'type',\n",

" 536: 'hard',\n",

" 537: 'frnd',\n",

" 538: 'boytoy',\n",

" 539: 'exam',\n",

" 540: 'secret',\n",

" 541: 'anytime',\n",

" 542: 'far',\n",

" 543: 'pmessage',\n",

" 544: 'mobileupd',\n",

" 545: 'dating',\n",

" 546: 'welcome',\n",

" 547: 'gd',\n",

" 548: 'party',\n",

" 549: 'crazy',\n",

" 550: 'wonderful',\n",

" 551: 'camcorder',\n",

" 552: 'used',\n",

" 553: 'gone',\n",

" 554: 'hit',\n",

" 555: 'operator',\n",

" 556: 'friday',\n",

" 557: 'mu',\n",

" 558: 'nt',\n",

" 559: 'ltd',\n",

" 560: 'wit',\n",

" 561: 'muz',\n",

" 562: 'carlos',\n",

" 563: 'finally',\n",

" 564: 'college',\n",

" 565: 'oredi',\n",

" 566: 'congratulation',\n",

" 567: 'read',\n",

" 568: 'light',\n",

" 569: 'project',\n",

" 570: 'fucking',\n",

" 571: 'visit',\n",

" 572: 'nope',\n",

" 573: 'outside',\n",

" 574: 'fri',\n",

" 575: 'pretty',\n",

" 576: 'sea',\n",

" 577: 'fast',\n",

" 578: 'drug',\n",

" 579: 'wonder',\n",

" 580: 'lovely',\n",

" 581: 'loving',\n",

" 582: 'wkly',\n",

" 583: 'info',\n",

" 584: 'hungry',\n",

" 585: 'seeing',\n",

" 586: 'telling',\n",

" 587: 'whole',\n",

" 588: 'frnds',\n",

" 589: 'dont',\n",

" 590: 'quiz',\n",

" 591: 'log',\n",

" 592: 'course',\n",

" 593: 'mrng',\n",

" 594: 'darlin',\n",

" 595: 'goodmorning',\n",

" 596: 'case',\n",

" 597: 'meant',\n",

" 598: 'cd',\n",

" 599: 'hold',\n",

" 600: 'unlimited',\n",

" 601: 'fone',\n",

" 602: 'jay',\n",

" 603: 'bout',\n",

" 604: 'ten',\n",

" 605: 'cum',\n",

" 606: 'term',\n",

" 607: 'listen',\n",

" 608: 'snow',\n",

" 609: 'gal',\n",

" 610: 'wrong',\n",

" 611: 'press',\n",

" 612: 'valued',\n",

" 613: 'im',\n",

" 614: 'child',\n",

" 615: 'almost',\n",

" 616: 'etc',\n",

" 617: 'move',\n",

" 618: 'cut',\n",

" 619: 'hee',\n",

" 620: 'download',\n",

" 621: 'computer',\n",

" 622: 'mah',\n",

" 623: 'bday',\n",

" 624: 'felt',\n",

" 625: 'invited',\n",

" 626: 'msg',\n",

" 627: 'joy',\n",

" 628: 'f',\n",

" 629: 'march',\n",

" 630: 'optout',\n",

" 631: 'side',\n",

" 632: 'tel',\n",

" 633: 'fr',\n",

" 634: 'dnt',\n",

" 635: 'single',\n",

" 636: 'bslvyl',\n",

" 637: 'lost',\n",

" 638: 'reading',\n",

" 639: 'txts',\n",

" 640: 'currently',\n",

" 641: 'moment',\n",

" 642: 'motorola',\n",

" 643: 'talking',\n",

" 644: 'couple',\n",

" 645: 'rock',\n",

" 646: 'as',\n",

" 647: 'different',\n",

" 648: 'le',\n",

" 649: 'savamob',\n",

" 650: 'sub',\n",

" 651: 'within',\n",

" 652: 'yar',\n",

" 653: 'happiness',\n",

" 654: 'area',\n",

" 655: 'paper',\n",

" 656: 'opt',\n",

" 657: 'sex',\n",

" 658: 'mayb',\n",

" 659: 'wats',\n",

" 660: 'least',\n",

" 661: 'earlier',\n",

" 662: 'film',\n",

" 663: 'luck',\n",

" 664: 'chennai',\n",

" 665: 'enter',\n",

" 666: 'gas',\n",

" 667: 'freemsg',\n",

" 668: 'promise',\n",

" 669: 'eh',\n",

" 670: 'national',\n",

" 671: 'eg',\n",

" 672: 'hospital',\n",

" 673: 'wow',\n",

" 674: 'correct',\n",

" 675: 'pas',\n",

" 676: 'complimentary',\n",

" 677: 'xx',\n",

" 678: 'ampm',\n",

" 679: 'load',\n",

" 680: 'shower',\n",

" 681: 'askd',\n",

" 682: 'sending',\n",

" 683: 'direct',\n",

" 684: 'tired',\n",

" 685: 'ni',\n",

" 686: 'mr',\n",

" 687: 'bill',\n",

" 688: 'semester',\n",

" 689: 'laptop',\n",

" 690: 'blue',\n",

" 691: 'swing',\n",

" 692: 'die',\n",

" 693: 'christmas',\n",

" 694: 'via',\n",

" 695: 'figure',\n",

" 696: 'ish',\n",

" 697: 'ago',\n",

" 698: 'chikku',\n",

" 699: 'seen',\n",

" 700: 'rental',\n",

" 701: 'supposed',\n",

" 702: 'park',\n",

" 703: 'ipod',\n",

" 704: 'gym',\n",

" 705: 'kid',\n",

" 706: 'darren',\n",

" 707: 'an',\n",

" 708: 'sell',\n",

" 709: 'eye',\n",

" 710: 'store',\n",

" 711: 'picking',\n",

" 712: 'ugh',\n",

" 713: 'extra',\n",

" 714: 'sort',\n",

" 715: 'knew',\n",

" 716: 'photo',\n",

" 717: 'support',\n",

" 718: 'information',\n",

" 719: 'surprise',\n",

" 720: 'grin',\n",

" 721: 'difficult',\n",

" 722: 'john',\n",

" 723: 'father',\n",

" 724: 'comp',\n",

" 725: 'usf',\n",

" 726: 'request',\n",

" 727: 'link',\n",

" 728: 'comin',\n",

" 729: 'cheer',\n",

" 730: 'charged',\n",

" 731: 'confirm',\n",

" 732: 'abiola',\n",

" 733: 'crave',\n",

" 734: 'checking',\n",

" 735: 'save',\n",

" 736: 'loan',\n",

" 737: 'page',\n",

" 738: 'entered',\n",

" 739: 'dogging',\n",

" 740: 'txting',\n",

" 741: 'lovable',\n",

" 742: 'wine',\n",

" 743: 'safe',\n",

" 744: 'bath',\n",

" 745: 'orchard',\n",

" 746: 'kate',\n",

" 747: 'lover',\n",

" 748: 'wana',\n",

" 749: 'hmv',\n",

" 750: 'somebody',\n",

" 751: 'return',\n",

" 752: 'small',\n",

" 753: 'discount',\n",

" 754: 'slow',\n",

" 755: 'mm',\n",

" 756: 'india',\n",

" 757: 'hmm',\n",

" 758: 'asking',\n",

" 759: 'rent',\n",

" 760: 'lady',\n",

" 761: 'door',\n",

" 762: 'train',\n",

" 763: 'laugh',\n",

" 764: 'idea',\n",

" 765: 'whenever',\n",

" 766: 'truth',\n",

" 767: 'heard',\n",

" 768: 'frm',\n",

" 769: 'fantasy',\n",

" 770: 'fact',\n",

" 771: 'loved',\n",

" 772: 'slowly',\n",

" 773: 'police',\n",

" 774: 'nah',\n",

" 775: 'callertune',\n",

" 776: 'copy',\n",

" 777: 'reward',\n",

" 778: 'wap',\n",

" 779: 'england',\n",

" 780: 'turn',\n",

" 781: 'catch',\n",

" 782: 'oops',\n",

" 783: 'reached',\n",

" 784: 'joke',\n",

" 785: 'situation',\n",

" 786: 'short',\n",

" 787: 'rply',\n",

" 788: 'representative',\n",

" 789: 'gave',\n",

" 790: 'men',\n",

" 791: 'del',\n",

" 792: 'spend',\n",

" 793: 'bathe',\n",

" 794: 'bcoz',\n",

" 795: 'teach',\n",

" 796: 'road',\n",

" 797: 'kick',\n",

" 798: 'admirer',\n",

" 799: 'deep',\n",

" 800: 'leaf',\n",

" 801: 'stupid',\n",

" 802: 'sim',\n",

" 803: 'somewhere',\n",

" 804: 'immediately',\n",

" 805: 'custcare',\n",

" 806: 'met',\n",

" 807: 'ex',\n",

" 808: 'woke',\n",

" 809: 'yep',\n",

" 810: 'voice',\n",

" 811: 'ldn',\n",

" 812: 'booked',\n",

" 813: 'remove',\n",

" 814: 'nyt',\n",

" 815: 'monday',\n",

" 816: 'water',\n",

" 817: 'opinion',\n",

" 818: 'member',\n",

" 819: 'hoping',\n",

" 820: 'across',\n",

" 821: 'warm',\n",

" 822: 'cheap',\n",

" 823: 'study',\n",

" 824: 'ho',\n",

" 825: 'gap',\n",

" 826: 'fantastic',\n",

" 827: 'glad',\n",

" 828: 'wishing',\n",

" 829: 'summer',\n",

" 830: 'gettin',\n",

" 831: 'understand',\n",

" 832: 'poor',\n",

" 833: 'otherwise',\n",

" 834: 'ntt',\n",

" 835: 'rd',\n",

" 836: 'na',\n",

" 837: 'doctor',\n",

" 838: 'energy',\n",

" 839: 'shes',\n",

" 840: 'write',\n",

" 841: 'cover',\n",

" 842: 'doin',\n",

" 843: 'med',\n",

" 844: 'empty',\n",

" 845: 'la',\n",

" 846: 'cup',\n",

" 847: 'std',\n",

" 848: 'seriously',\n",

" 849: 'suck',\n",

" 850: 'worried',\n",

" 851: 'sick',\n",

" 852: 'decide',\n",

" 853: 'forever',\n",

" 854: 'rain',\n",

" 855: 'bos',\n",

" 856: 'specially',\n",

" 857: 'flight',\n",

" 858: 'ending',\n",

" 859: 'buying',\n",

" 860: 'sunshine',\n",

" 861: 'sony',\n",

" 862: 'lazy',\n",

" 863: 'lect',\n",

" 864: 'completely',\n",

" 865: 'hmmm',\n",

" 866: 'staying',\n",

" 867: 'spoke',\n",

" 868: 'especially',\n",

" 869: 'studying',\n",

" 870: 'trust',\n",

" 871: 'using',\n",

" 872: 'deal',\n",

" 873: 'mrt',\n",

" 874: 'disturb',\n",

" 875: 'unless',\n",

" 876: 'rose',\n",

" 877: 'sport',\n",

" 878: 'accept',\n",

" 879: 'normal',\n",

" 880: 'rest',\n",

" 881: 'merry',\n",

" 882: 'pete',\n",

" 883: 'access',\n",

" 884: 'round',\n",

" 885: 'urself',\n",

" 886: 'bluetooth',\n",

" 887: 'reference',\n",

" 888: 'brings',\n",

" 889: 'mistake',\n",

" 890: 'result',\n",

" 891: 'style',\n",

" 892: 'meh',\n",

" 893: 'near',\n",

" 894: 'silent',\n",

" 895: 'hotel',\n",

" 896: 'noon',\n",

" 897: 'forwarded',\n",

" 898: 'starting',\n",

" 899: 'xy',\n",

" 900: 'bold',\n",

" 901: 'seems',\n",

" 902: 'add',\n",

" 903: 'wwwgetzedcouk',\n",

" 904: 'bid',\n",

" 905: 'charity',\n",

" 906: 'fall',\n",

" 907: 'tampa',\n",

" 908: 'user',\n",

" 909: 'sale',\n",

" 910: 'gay',\n",

" 911: 'hiya',\n",

" 912: 'convey',\n",

" 913: 'digital',\n",

" 914: 'nobody',\n",

" 915: 'mode',\n",

" 916: 'wondering',\n",

" 917: 'others',\n",

" 918: 'bb',\n",

" 919: 'frens',\n",

" 920: 'tht',\n",

" 921: 'reaching',\n",

" 922: 'didnt',\n",

" 923: 'excuse',\n",

" 924: 'thinkin',\n",

" 925: 'sitting',\n",

" 926: 'flag',\n",

" 927: 'colleague',\n",

" 928: 'entitled',\n",

" 929: 'anymore',\n",

" 930: 'hl',\n",

" 931: 'sunday',\n",

" 932: 'mark',\n",

" 933: 'pizza',\n",

" 934: 'quick',\n",

" 935: 'replying',\n",

" 936: 'roommate',\n",

" 937: 'letter',\n",

" 938: 'nigeria',\n",

" 939: 'ice',\n",

" 940: 'cinema',\n",

" 941: 'stand',\n",

" 942: 'spent',\n",

" 943: 'trouble',\n",

" 944: 'planning',\n",

" 945: 'ave',\n",

" 946: 'apartment',\n",

" 947: 'inc',\n",

" 948: 'paying',\n",

" 949: 'bak',\n",

" 950: 'dvd',\n",

" 951: 'pray',\n",

" 952: 'sometimes',\n",

" 953: 'goto',\n",

" 954: 'freephone',\n",

" 955: 'however',\n",

" 956: 'dead',\n",

" 957: 'slept',\n",

" 958: 'gt',\n",

" 959: 'file',\n",

" 960: 'sign',\n",

" 961: 'street',\n",

" 962: 'youfind',\n",

" 963: 'rreveal',\n",

" 964: 'specialcall',\n",

" 965: 'goodnight',\n",

" 966: 'power',\n",

" 967: 'record',\n",

" 968: 'woman',\n",

" 969: 'fixed',\n",

" 970: 'wiv',\n",

" 971: 'weed',\n",

" 972: 'hgsuitelands',\n",

" 973: 'rowwjhl',\n",

" 974: 'mei',\n",

" 975: 'picture',\n",

" 976: 'facebook',\n",

" 977: 'none',\n",

" 978: 'yahoo',\n",

" 979: 'aha',\n",

" 980: 'funny',\n",

" 981: 'giving',\n",

" 982: 'din',\n",

" 983: 'mp',\n",

" 984: 'fat',\n",

" 985: 'finger',\n",

" 986: 'self',\n",

" 987: 'workin',\n",

" 988: 'daddy',\n",

```
        " 989: 'present',\n",
        " 990: 'indian',\n",
        " 991: 'euro',\n",
        " 992: 'future',\n",
        " 993: 'longer',\n",
        " 994: 'pc',\n",
        " 995: 'tuesday',\n",
        " 996: 'student',\n",
        " 997: 'tonite',\n",
        " 998: 'library',\n",
        " 999: 'space',\n",
        " 1000: 'slave',\n",
        " ...}"
       ]
      },
      "metadata": {},
      "execution_count": 55
    }
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "from keras_preprocessing.sequence import pad_sequences"
  ],
  "metadata": {
   "id": "rmxJsoYd_r5G"
  },
  "execution_count": 75,
  "outputs": []
 },
```

```
{
 "cell_type": "code",
 "source": [
  "max_length_sequence = max([len(i) for i in text_to_sequence])\n",
  " # finding the length of largest sequence\n",
  "padded_sms_sequence = pad_sequences(text_to_sequence, maxlen=max_length_sequence, padding=\"pre\") \n",
  "padded_sms_sequence"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "Nk2VDdmX6PDc",
  "outputId": "6151eac8-153b-4b78-9171-eddc90c8334c"
 },
 "execution_count": 76,
 "outputs": [
  {
   "output_type": "execute_result",
   "data": {
    "text/plain": [
     "array([[   0,    0,    0, ...,    9, 3576,   70],\n",
     "       [   0,    0,    0, ..., 1173,  328, 1539],\n",
     "       [   0,    0,    0, ..., 2413,  263, 2414],\n",
     "       ...,\n",
     "       [   0,    0,    0, ..., 1024, 7818, 3573],\n",
     "       [   0,    0,    0, ...,  789,   67,    5],\n",
     "       [   0,    0,    0, ..., 2147,  363,  145]], dtype=int32)"
    ]
   },
```

```
      "metadata": {},
      "execution_count": 76
     }
    ]
   },
   {
    "cell_type": "code",
    "source": [
     "VOC_SIZE = len(tokenizer.word_index)+1\n",
     "def create_model():\n",
     "    \n",
     "     model = Sequential()\n",
     "     model.add(Embedding(VOC_SIZE, 32, input_length=max_length_sequence))\n",
     "     model.add(LSTM(100))\n",
     "     model.add(Dropout(0.4))\n",
     "     model.add(Dense(20, activation=\"relu\"))\n",
     "     model.add(Dropout(0.5))\n",
     "     model.add(Dense(1, activation = \"sigmoid\"))\n",
     "     return model\n",
     "lstm_model = create_model()\n",
     "lstm_model.compile(optimizer = \"adam\", loss = \"binary_crossentropy\", metrics = [\"accuracy\"])\n",
     "lstm_model.summary()"
    ],
    "metadata": {
     "colab": {
      "base_uri": "https://localhost:8080/"
     },
     "id": "VxfhKVjm6zwD",
     "outputId": "7de4d46d-e67d-4505-f28a-0c33bd42871a"
    },
```

```
 "execution_count": 77,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "Model: \"sequential\"\n",
    "_________________________________________________________________\n",
    " Layer (type)                Output Shape              Param #   \n",
    "=================================================================\n",
    " embedding (Embedding)       (None, 79, 32)            250240    \n",
    "                                                                 \n",
    " lstm (LSTM)                 (None, 100)               53200     \n",
    "                                                                 \n",
    " dropout (Dropout)           (None, 100)               0         \n",
    "                                                                 \n",
    " dense (Dense)               (None, 20)                2020      \n",
    "                                                                 \n",
    " dropout_1 (Dropout)         (None, 20)                0         \n",
    "                                                                 \n",
    " dense_1 (Dense)             (None, 1)                 21        \n",
    "                                                                 \n",
    "=================================================================\n",
    "Total params: 305,481\n",
    "Trainable params: 305,481\n",
    "Non-trainable params: 0\n",
    "_________________________________________________________________\n"
   ]
  }
 ]
},
```

```
{
 "cell_type": "code",
 "source": [
  "lstm_model.fit(padded_sms_sequence, y, epochs = 5, batch_size=16, validation_split=0.2)"
 ],
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "MU6oJZQCAF-E",
  "outputId": "dc475e7e-bbd7-4908-b69a-cb7064fbb929"
 },
 "execution_count": 79,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "Epoch 1/5\n",
    "279/279 [==============================] - 20s 62ms/step - loss: 0.1865 - accuracy: 0.9437 - val_loss: 0.0587 - val_accuracy: 0.9803\n",
    "Epoch 2/5\n",
    "279/279 [==============================] - 17s 60ms/step - loss: 0.0485 - accuracy: 0.9910 - val_loss: 0.0862 - val_accuracy: 0.9731\n",
    "Epoch 3/5\n",
    "279/279 [==============================] - 17s 60ms/step - loss: 0.0250 - accuracy: 0.9951 - val_loss: 0.0791 - val_accuracy: 0.9785\n",
    "Epoch 4/5\n",
    "279/279 [==============================] - 19s 67ms/step - loss: 0.0069 - accuracy: 0.9991 - val_loss: 0.1132 - val_accuracy: 0.9839\n",
    "Epoch 5/5\n",
    "279/279 [==============================] - 19s 67ms/step - loss: 0.0077 - accuracy: 0.9987 - val_loss: 0.1186 - val_accuracy: 0.9803\n"
```

```
    ]
   },
   {
    "output_type": "execute_result",
    "data": {
     "text/plain": [
      "<keras.callbacks.History at 0x7f9ec2f92a10>"
     ]
    },
    "metadata": {},
    "execution_count": 79
   }
  ]
 },
 {
  "cell_type": "code",
  "source": [
   "pickle.dump(tokenizer, open(\"sms_spam_tokenizer.pkl\", \"wb\"))\n",
   "pickle.dump(lstm_model, open(\"lstm_model.pkl\", \"wb\"))"
  ],
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "4VkZFIpCAxmT",
   "outputId": "74214afe-807f-4b23-eb77-397da490d4c8"
  },
  "execution_count": 80,
  "outputs": [
   {
    "output_type": "stream",
```

```
        "name": "stderr",
        "text": [
          "WARNING:absl:Found untraced functions such as lstm_cell_layer_call_fn, lstm_cell_layer_call_and_return_conditional_losses while saving (showing 2 of 2). These functions will not be directly callable after loading.\n"
        ]
      }
    ]
  }
 ]
}
```