ASSIGNMENT 4

```
{
    "nbformat": 4,
    "nbformat_minor": 0,
    "metadata": {
    "colab": {
    "provenance": [],
    "collapsed_sections": []
    },
    "kernelspec": {
    "name": "python3",
    "display_name": "Python 3"
    },
    "language_info": {
    "name": "python"
    }
    },
      "cells": [
        {
          "cell_type": "markdown",
          "source": [
            "**Importing** **the** **libraries**"
          ],
          "metadata": {
            "id": "wN7qBcvxaC_z"
          }
        },
        {
          "cell_type": "code",
          "source": [
            "from google.colab import drive\n",
            "drive.mount('/content/drive')"
          ],
          "metadata": {
            "colab": {
              "base_uri": "https://localhost:8080/"
            },
            "id": "nadwGSmE4q7i",
            "outputId": "90ef124f-99a0-4430-8e79-6cf441aede02"
          },
          "execution_count": 3,
          "outputs": [
            {
              "output_type": "stream",
              "name": "stdout",
          "text": [
            "Mounted at /content/drive\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "import pandas as pd\n",
        "import numpy as np\n",
        "import matplotlib.pyplot as plt\n",
        "import seaborn as sns\n",
        "from sklearn.model_selection import train_test_split\n",
        "from sklearn.preprocessing import LabelEncoder\n",
        "from keras.models import Model,Sequential\n",
        "from keras.layers import LSTM, Activation, Dense, Dropout, Input,
Embedding\n",
        "from keras.optimizers import Adam\n",
        "from keras.preprocessing.text import Tokenizer\n",
        "from keras.preprocessing import sequence\n",
        "from keras.utils import pad_sequences\n",
        "from keras.utils import to_categorical\n",
        "from keras.callbacks import EarlyStopping\n",
        "%matplotlib inline\n"
      ],
      "metadata": {
        "id": "_plGwx1aaMIg"
      },
      "execution_count": null,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "**Reading** **the** **dataset**"
      ],
      "metadata": {
        "id": "PnMYl7rza2dK"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "df = pd.read_csv('/content/drive/MyDrive/spam.csv',encoding='latin-1')\n",
        "df.head()\n"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 206
        },
        "id": "iR6LXy-za-FY",
        "outputId": "5ee64950-26c2-4897-b9f0-31baa1b52d13"
      },
      "execution_count": 4,
      "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-3a5fb1c0-e699-4076-b730-9c9ad6f1eca4\">\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-3a5fb1c0-e699-4076-b730-9c9ad6f1eca4')\"\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-3a5fb1c0-e699-4076-b730-
9c9ad6f1eca4 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-3a5fb1c0-e699-
4076-b730-9c9ad6f1eca4');\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": 4
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed:
4'],axis=1,inplace=True)\n",
        "df.info()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "VdVbdZHcbYgO",
        "outputId": "29ebcdb5-7f2c-4ec4-b191-0c6367d5821e"
      },
      "execution_count": 5,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "<class 'pandas.core.frame.DataFrame'>\n",
            "RangeIndex: 5572 entries, 0 to 5571\n",
            "Data columns (total 2 columns):\n",
            " #   Column  Non-Null Count  Dtype \n",
            "---  ------  --------------  ----- \n",
            " 0   v1      5572 non-null   object\n",
            " 1   v2      5572 non-null   object\n",
            "dtypes: object(2)\n",
            "memory usage: 87.2+ KB\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "sns.countplot(df.v1)\n",
        "plt.xlabel('Label')\n",
        "plt.title('Number of ham and spam messages')\n"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 367
        },
        "id": "1oCcVv_2baSN",
        "outputId": "ff38066e-0d96-410a-d8ce-51d9ee65d0aa"
      },
      "execution_count": 6,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stderr",
          "text": [
            "/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12,
the only valid positional argument will be `data`, and passing other arguments
without an explicit keyword will result in an error or misinterpretation.\n",
            "  FutureWarning\n"
          ]
        },
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "Text(0.5, 1.0, 'Number of ham and spam messages')"
            ]
          },
          "metadata": {},
     "execution_count": 6
    },
    {
     "output_type": "display_data",
     "data": {
      "text/plain": [
       "<Figure size 432x288 with 1 Axes>"
      ],
      "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYsAAAEWCAYAAACXGLsWAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAALEg
AACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsIGh0dHA6Ly9tYXRwb
G90bGliLm9yZy+WH4yJAAAZjUlEQVR4nO3deZhddZ3n8feHBEQFJUhESGjDKLbirhGwtbtpHAFxgcd2wXEJ
imI72toz7do9I4j4qK0j7nbTioCOIu5p2xZRcRtFSxAQCXDYsKWAKoiPwnT/Or+RQVOVOVILeqknq/nuc
+dc7vLPd7zr11P/esN1WFJEkbs81MFyBJmv0MC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQlOS5KQkx83Qcy
fJR5NsSHL2BMOPSPPLdmahtc0qyf5I1M12HNBHDYguV5NIka5Pcvdf24iTfnMGyRuXxwBOBxVW1z0wXI81Fh
sWWbR7wqpkuYlMlmbeJi9wXuL6qft1FPW2xJFtHiq0NOMKeiQNYMy2byQ0KNMyy2bO8AXp1kp/EDkixJUknm99q+meTFrfuIP8Qcy
fJR5NsSHL2BMOPSPLdmahtc0qyf5I1M12HNBHDYguV5NIka5Pcvdf24iTfnMGyRuXxwBOBxVW1z0wXI81Fh
sWWbR7wqpkuYlMlmbeJk9wXuLSqfjOKeiQNMyy2bO8AXp1kp/EDkixJUknm99q+meTFrfuIJP8nyfFJrk1y
cZI/a+2r21bLsnGz3SXJGUmuT/KtJPftzT/KQ3vr95428NuSfCjCjJl5P8BvirCerdPcnyNv2qJC9p7Uc
CHwYem+SGJG+abGUkeWfbVXJkif12l+Y5MJW98VJXtobtn+SNb+yiSHJTkkySkS9aPf+wked8cpIfJf
l1W2/HTPAaLEvyyyyS/SvKPveF3betmQ5ILbMds5HnSXqu17bnOS/KQ3vr95428Nu9ptf06ycokf94bdkyST
yf5eJv2vCQPPSPKG9lyrkxy4kouTfKaJOcm+U2Sjyr/tPl9LcmC3vjLee8Rdk62DOaOqfGyBD+BS4D8
DnwOOa20vBBr7ZupcABczvTfNN4MWt+wjgJuCFdFsoxwG/BD4A3A0d9uwuwOr27
zmA48EfgXs3Zv2OuBxdF9Qtp9geb4NfBDYHngEsA44oFfrdzeyLo4A/gC8pFdxGKdSYVf8ZQYBoDrL4E3gEMvNe0ko2IoL5dFKwUBHfLUCGe3H6QzXhp9G
KPasP3bengjsG2bxzrgE8COwIOBG4E9J3nu/YGHtuV6GHA1cNi41+BfgbsCDwd+k41HDaxHd/FgJE0c39ifc
mkme5yBgJbBTW44HAbsNvTZt+POAe7XX5u+Bq8ZeA+AY4zOzAouAtcAP23TtYmH/srY9LBgH/srY9LBt6HZwG7AouAtcAP23t
ge+AbwNFt3EXANcAhbX09sfUvpHsP/Rr40zbubsCDW/cnWz3btHk4+forL9jbgW8ACYDFw7tj6bfNa2V6YCD/Vz7Y737Y
D/BFwMHNSGfx94fuveAdvpv/nZ/ox4wX4uIMv3K1h8RC6D+KFbHpYXNQb9tA2/q69tmuAR7Tuk4BPN8N2A
G6m+5B7NvCdcsmeEgI0syx5tXjv22t4KnNSrdSgsbG1nlsVVvX679aaW5+3Gb5VR5jC29M2a+7b
G38lLQCm8Lq8Gzh+3Guwu8bjZuTd4BxhT5EN9eb0pbQO8cGo5vkrvwfd2OfqcqSnvhk3s2i2m9rsBdGA=m31bN4T9eE9PrcAv8fcE/nkrT5n0lvXVK
rY6eNvA+f2+v/LPChXv/fAl9o3a8DPjZu+tOBZXRhcS3w18Bdx41zrCnBCfz1uZP33l+2PH/6t/8XcGhb7Ar
8cN+0bgI+27m8DbwJ22Zz/t1vyw91QW7iq+inwJeD9Pid3t+kY/DWrJrpajDgd4ENeQWhDaHSpfcTkeZ8B2hQbWRbKFtC0/ebDQjJ9fFu3hO4A9DWFe7Sq2bo6VmhQ1LJWTAfRTA7ZuNhUm
pIAkQxqgIYQwA5cT2XfyMZ31b2WX6dD6ZfLyVXRTchymCYuWRz/bxQhcHfw4iCLB+B1TlyTGnX2v8gDEWH5IyN
bqAS0Ga9mxt4tgumm0t7frgvQ2KmqASeDvVCW5VgsJm3+H6R4mN20Zr/OPaxdZ1D/nm5rG7DEfa6G5H58KfpkxeUaZXT5z3CAzGEOj1nb2xW0gsEYiE
sP8BDs3/l5EbBbKNUbFBSnJq7b6brc1c4qQ/zFz76dZT4Kxv2ZFVwhVtxWpxX7vp6mG8SuXHvwCeh/cEuYDN2rEUqTJ4bxTsVRFubSKpkMPhP+Ru
2tBZSZBYmkTzLhAAzbhd2JVGQJGrn2M4pwyN6ZvQmv1cIV2/4NMKftdmSq+4g5HcjbFdMZShV7Ubru/Nmjr8X3
yV6bHemOyawD5id5I3APZsbHgzcmOai9J7dPd4LB4nZQ/ND2gfx7ul1htwAkeWZbP9DtZqo2bGjZ+ut3EfC
K3rCzgeuTvK4dCJ+X5CFJHtOe83Lzm7x+C263vOMSy2HsfS7fftewndt89r6A7Ufu9OPsncn6LZi1g
OPpju4SNt9dCBwON1WwlV038busgnzfg7dPv4rgM/THe/42p2sd6y2V9J9cGyg+zBffmfn2/NfgWOTXE93s
PS0TZj2TXS7ni4Bvgp8bCPj2TXS7ni4Bvgp8bCPj3oPum/aGNs01dGfDjZnwtaE7JvAVuuMdl9GFzMZ2CY5MC69D6Xb3rGt1vIbu
c2gb4L/Tvf7r6U5EGNsyfQzwgyQ30L12r6qixletmOBNXTr92vAZ+iCiLab7Sl0J1NcQndCxofpdmMBHAy
c357zPXTHmW5kDhs7W0TSFirJSXQHbv/HTNcymyV5Gd2H/l/OdC1bIrcsJG2VkuyqsU24u/Q0ub0AW4
c357zPXTHmW5kDhs7W0TSFirJSXQHbv/HTNcymyV5Gd2H/l/OdC1bIrcsJG2VkuyW5HFJtknyp3Sn1n5+pu
```

vaUs0fHkWStkjb0Z3CvSfdcYdT6a7l0R3gbihJ0iB3Q0mSBo10N1SSS+luQ3AzcFNVLU2yM91pnkvorv58V
lVtSBK6sw4Oobtw6Yiq+mGbzzJg7ODdcVV18saed5dddqklS5Zs9uWRpK3ZypUrf1VVCycaNh3HLP6qqn7V
63898PWqeluS17f+1wFPAvZqj32BD9FdFbwz3SmBS+nOr16ZZHlVbZjsCZcsWcKKFStGszSStJVKMuldBGZ
iN9ShwNiWwcnAYb32U6pzFrBTkt3obnB2RlWtbwFxBt050JKkaTLqsCjgq+3WwUe1tl2r6srWfRW33otlEb
e9oGZNa5us/TaSHNVuKbxi3bp1m3MZJGnOG/VuqMdX1eVJ7g2ckeRn/YFVVUk2y+lYVXUC3Z0pWbp0qad4S
dJmNNIti6q6vP1dS3cxzD7A1W33Eu3v2jb65dz2Rl+LW9tk7ZKkaTKysEhy97G7iLabgx1I9+Muy+nuX0/7
+8XWvRx4QTr7Ade13VWnAwe2m4EtaPM5fVR1S5Jub5S7oXYFPt+dEct84BNV9ZUk5wCnpfu5zMvobi8M3W2
jDwFW0Z06+0KAqlqf5M3AOW28Y6tq/QjrliSNs1Vewb106dLy1FlJ2jRJVlbV0omGeQW3JGmQYSFJGuRdZy
fx6NecMtMlaBZa+Y4XzHQJ0oxwy0KSNMiwkCQNMiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0y
LCQJA0yLCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDDAtJ0iDDQpI0yLCQJA0y
LCRJgwwLSdIgw0KSNMiwkCQNMiwkSYMMC0nSIMNCkjTIsJAkDTIsJEmDRh4WSeYl+VGSL7X+PZP8IMmqJJ9
Ksl1rv0vrX9WGL+nN4w2t/edJDhp1zZKk25qOLYtXARf2+t8OHF9V9wc2AEe29iOBDa39+DYeSfYGDgceDB
wMfDDJvGmoW5LUjDQskiwGngx8uPUHOAD4TBvlZOCw1n1o66cNf0Ib/1Dg1Kr6fVVdAqwC9hll3ZKk2xr1l
sW7gdcCt7T+ewHXVtVNrX8NsKh1LwJWA7Th17Xx9/9g+wTR/lOSoJCuSrFi3bt3mXg5JmtNGFhZJngKsraqV
o3qOvqo6oaqWVtXShQsXTsdTStKcMX+E834c8LQkhwDbA/cA3gxGR+23pYDFzexr8c2ANYk2Q+cE/gml7
7mP40kqRpMLIti6p6Q1UtrqoldAeov1FVzwXOBJ7RBJ7RRlsGfLF1L2/9tOHfqKpq7Ye3s6X2BPYCzh5V3ZKk2x
vllsVkXgecmuQ44EfAR1r7R4CPJVkFrKcLGKrq/CSnARuBBA/cA3gPslGR+23pYDFzexr8c2ANYk2Q+cE/gml7
7mP40kqRpMLIti6p6Q1UtrqoldAeov1FVzwXOBJ7RRlsGfLF1L2/9tOHfqKpq7Ye3s6X2BPYCzh5V3ZKk2x
vllsVkXgecmuQ44EfAR1r7R4CPJVkFrKcLGKrq/CSnARcANwEvr6qbp79sSZq7piUsquqbwDdb98VMcDZTV
f0OeOYk078FeMvoKpQkbYxXcEuSBhkWkqRBhoUkaZBhIUkaZFhIkgYZFpKkQYaFJGmQYSFJGmRYSJIGGRaS
pEGGhSRpkGEhSRpkWEiSBhkWkqRBhoUkaZBhIUkaZFhIkgYZFpKkQYaFJGmQYSFJGmRYSJIGGRaSpEGGhSR
pkGEhSRpkWEiSBhkWkqRBhoUkaZBhIUkaZFhIkgYZFpKkQYaFJGmQYSFJGmRYSJIGGRaSpEEjC4sk2yc5O8
lPkpyf5E2tfc8kZyVZleRTSbZr7Xdr/ava8CW9eb2htV+Y5MkjWyhJ0qTYsjWadVYtMqVvAN+6bxmkiNQ21w
1fV/YENwJFt/COBDa39+DYeSfYGDgceDBwMfDDJvBHWLUkaZ2RhUZ0bVv+bu+27VHAAcBnMvvvJwGGt+9DWTxv+
hCRp7adW1e+r6hJgFbDPqAWJN3eSI9ZJJmX5MfAWuAM4M4P8C11bVTW2UNcCi1r0IWA3Qhl8H3KvfPsE0/ec
6KsmKJCvWrVs3isWDlrpGFRVTdX1SOAxXRbAw8c4XOdUFVLq2rpwoULp/QkjQnTcvZUFV1LXm/Am8Fhgpy
Tz26DFwOWt+3JgDA2/J7J7J7ANf32CaSRJE2DUZ4MtTLJz634B8CRgJd0ZrxGtvsBnnUnf9LOUu4NmM+JCc
xMWObQmbUd1J3gePE7P3YNPFudFqlFRSHLZ/FwpuxRd3v+OkjQNiyNWZ3FTCKaJvUPGAkX3ztMfGwmzALtqunnqX+23yVGEXMOeudbAPdgfS9ANqc4
HshDIDwKiQRcQ0N/SPZz0HIQ0NsUdY0M+1V2XJIRaVQJiD+X6hfMhM2Awm+3EwwQN7hb4Nb8PygaktuvdcEmOmBQ7rT+C
k+fpU2iRJW6eNHuBOsj1wN2CXJAuAtEH3YIKrqCVJW6ehs6FeCvwJ9G5EVvzkNdYBvwHdrVR4sWpBd0ooQYzn1nd
D3JPnbqnrfNNUkSZplpnSdRW9L8mfuFtq0wpBLJJ8DLgf8GNg7OrpAgwLSZoDpoRsnVuvv
nFjq5P/Qe1Co14PUsNoBpmKaUMtQ1Hje6Uh6cn+rJXeWtQo3nJhIm4FcCTdRrpP5Kx2Pz/tjdVOTwoHCQdx6pdLtH4wnUFjAjqUf4rBE1RtwP1rR1UwUWjsNhO/Cg3yRcgyDqLpI8Dh9fBpFRRgdaxIZvkgyPdr7KxjbdDMQ7KgX1UHB/wKWaZOMBrN71I0I1Sn1dKFzFn6fbnoHmQcoGs9MGrxMCVOk9h2OXuAKLQtRFB4aJN5WAo2Be+BqH1V2P7EL4DgBOVrVcIb5ZOmwoF2HMQIqE9q3TDrYv5iUMqeY4cODIbR

```
o4F9gX2Ao8cCRpI0fUYWFlX1bWD9uOZDgZNb98nAYb32U6pzFrBTkt2Ag4Azqmp9VW0AzuD2ASRJGrHpPma
xa1Vd2bqvAnZt3YuA1b3x1rS2ydpvJ8lRSVYkWbFu3brNW7UkzXEzdoC7qgqozTi/E6pqaVUtXbhw4earrS
SJ6Q+Lq9vuJdrfta39cmCP3niLW9tk7ZKkaTTdYbEcGDujaRnwxV77C9pZUfsB17XdVacDByZZ0A5sH9jaJ
EnTaP6oZpzkk8D+wC5J1tCd1fQ24LQkRwKXAc9qo38ZOARYBfwWeCFAVa1P8mbgnDbesVU1/qC5JGnERhYW
VfWcSQY9YYJxC3j5JPM5EThxM5YmSdpEXsEtSRpkWEiSBhkWkqRBhoUkaZBhIUkaNLKzoSSNxi+PfehW
Z6E/eeN5I5++WhSRpkGEhSRpkWEiSBhkWkqRBhoUkaZBhIUkaZFhIkgYZFpKkQYaFJGmQYSFJGmRYSJIGGR
aSpEGGhSRpkGEhSRpkWEiSBhkWkqRBhoUkaZBhIUkaZFhIkgYZFpKkQYaFJGmQYSFJGmRYSJIGGRaSpEGGh
SRpkGEhSRpkWEiSBhkWkqRBW0xYJDk4yc+TrEry+pmuR5Lmki0iLJLMAz4APAAnYG3hOkr1ntipJmju2iLAA
9gFWVdXFVfX/gFOBQ2e4JkmaM+bPdAFTtAhY3etfA+zbHyJUcBRrfeGJD+fptrmgl2AX810EbNB3rlspkv
QbfneHHN0Nsdc7jvZgC0lLAZV1QnACTNdx9YoyYqqWjrTdUjj+d6cPlvKbqjLgT16/YtbmyRpGmwpYXEOsF
eSPZNsBxwOLJ/hmiRpztgidkNV1U1JXgGcDswDTqyq82e4rLnE3XuarXxvTpNU1UzXIEma5baU3VCSpBlkW
EiSBhkWc1iSJUl+OtN1SJr9DAtJ0iDDQvOS/GuS85N8Ncldk7wkyTlJfpLks0nuBpDkCQfSnJWkouT7J/k
xCQXJjlphpdDW7gkd0/y7+1999Mkz05yaZJ/SnJekrOT3L+N+9QkP0jjyoyRfS7Jraz8myclJvpPksiRP703
/lSTbzuxSbrkMC+0FfKCqHgxcC/w18Lmqekxy4EDiyN/4C4LHAf6O71uV44MHAQ5M8Ylor19bmYOCKqn
p4VT0E+Eprv66qHgq8H3h3a/susF9VPZLuXnGv7c3nfsABwBwNOAjwNntulvBJ48+sXYOhkWuqSqfty6VwJLg
Ie0b2bnAc+lC4Mx/1bd+dbnAVdX1XlVdQtwfptWuqPOA56Y5O1J/ryqbgyc3t6jr+G279H/
qKo/tPnN49bQOQ/fo3eYYaHf97pvprtQ8yTgFe3b2JuA7ScY/5Zx097CFnCEJ6VmsabJ1/6KUMrWG0kkIjWuw
/tPb3fcD723v0pUzwHm1fYv5Qt15M5nv0TjAsNJEdgSvb/t3nznQxmhuS7A78tqo+DryDLjgAnt37+v3W/fU
9uvT+ctwKeBqasJvI/gR8A69rfHWe2HM0RDwXekeQW4A/Ay4DDXPAAuSnEu3xfCcNu4xwKeTbAC+Aew5/eXOL
d7uQ9KsleRSYGlV+ZsVM8zdUJKkQW5ZSJIGGuWUhSRpkWEiSBhkWkqRBhoV0JyS5YRPGGPGSbJq0c1f2mUDAtJ
0iDDQtrMJrsjavPwJN9PclGSl/SmeU270O/nSd7U2l/aNoNfzi/u6I+jC6O6I+Fnhjkt2THEh39999gEc
Aj07yF9Ncs7RR3u5D2vwWA59KshuwHXBJb9gXq+pG4MYkZ9IFxAuAuPb09fydLGGRbS5vc+4F
1VtTzJ/nT3MRoz/irYAgK8tar+ZXrKkzadu6GkzW9jd0Q9NMn2Se4F7A+cA5wOvCjJDgBJFiW593QVK02FW
xbSnXO3JGt6/e9i43dEPRc4E9gFeHNVXQFckeTANwAPA9YO/rypanx3lCSpEHuhpIkDTIsJEmDDAtJ
0iDDQpI0yLCQJA0yLCRJgwwLSdKg/w+zxO5JjPQAtAAAAABJRU5ErkJggg==\n"
```

```
          },
          "metadata": {
            "needs_background": "light"
          }
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "X = df.v2\n",
        "Y = df.v1\n",
        "le = LabelEncoder()\n",
        "Y = le.fit_transform(Y)\n",
        "Y = Y.reshape(-1,1)"
      ],
      "metadata": {
        "id": "Df4Z9022bh6-"
      },
      "execution_count": 7,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.15)\n"
      ],
      "metadata": {
        "id": "Nyesd4EabxuM"
      },
      "execution_count": 8,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "**Process the data**"
      ],
      "metadata": {
        "id": "FRuvmIAbj2Fz"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "max_words = 1000\n",
        "max_len = 150\n",
        "tok = Tokenizer(num_words=max_words)\n",
        "tok.fit_on_texts(X_train)\n",
        "sequences = tok.texts_to_sequences(X_train)\n",
        "sequences_matrix = pad_sequences(sequences,maxlen=max_len)\n",
        "sequences_matrix"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "x8vSHWa9bz1d",
        "outputId": "ea4d69ce-601d-43fb-bf2a-be3827efe0c8"
      },
      "execution_count": 9,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "array([[  0,   0,   0, ..., 260,  13, 469],\n",
          "       [  0,   0,   0, ...,   1,  66,   3],\n",
          "       [  0,   0,   0, ...,   1, 287, 544],\n",
          "       ...,\n",
          "       [  0,   0,   0, ...,  80,   2, 271],\n",
          "       [  0,   0,   0, ...,   0, 243,  11],\n",
          "       [  0,   0,   0, ...,  64,   6,  48]], dtype=int32)"
        ]
      },
      "metadata": {},
      "execution_count": 9
    }
  ]
},
{
  "cell_type": "code",
  "source": [
    "tok.index_word"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "9ldXOnM_c3eQ",
    "outputId": "594ccc76-f901-4f42-ac5a-6ec35c721a8b"
  },
  "execution_count": 10,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "{1: 'i',\n",
          " 2: 'to',\n",
          " 3: 'you',\n",
          " 4: 'a',\n",
          " 5: 'the',\n",
          " 6: 'u',\n",
          " 7: 'and',\n",
          " 8: 'in',\n",
          " 9: 'is',\n",
          " 10: 'me',\n",
          " 11: 'my',\n",
          " 12: 'for',\n",
          " 13: 'your',\n",
          " 14: 'it',\n",
          " 15: 'of',\n",
" 16: 'call',\n",
" 17: 'have',\n",
" 18: 'on',\n",
" 19: 'that',\n",
" 20: '2',\n",
" 21: 'now',\n",
" 22: 'are',\n",
" 23: 'but',\n",
" 24: 'not',\n",
" 25: 'so',\n",
" 26: 'or',\n",
" 27: 'can',\n",
" 28: 'do',\n",
" 29: 'at',\n",
" 30: \"i'm\",\n",
" 31: 'if',\n",
" 32: 'get',\n",
" 33: 'be',\n",
" 34: 'will',\n",
" 35: 'ur',\n",
" 36: 'no',\n",
" 37: 'just',\n",
" 38: 'with',\n",
" 39: 'we',\n",
" 40: 'this',\n",
" 41: 'gt',\n",
" 42: 'lt',\n",
" 43: '4',\n",
" 44: 'up',\n",
" 45: 'when',\n",
" 46: 'from',\n",
" 47: 'ok',\n",
" 48: 'free',\n",
" 49: 'out',\n",
" 50: 'all',\n",
" 51: 'what',\n",
" 52: 'how',\n",
" 53: 'go',\n",
" 54: 'know',\n",
" 55: 'like',\n",
" 56: 'then',\n",
" 57: 'got',\n",
" 58: 'good',\n",
" 59: 'was',\n",
" 60: 'time',\n",
" 61: 'come',\n",
" 62: 'only',\n",
" 63: 'am',\n",
" 64: 'day',\n",
" 65: 'its',\n",
" 66: 'love',\n",
" 67: 'there',\n",
" 68: 'he',\n",
" 69: 'text',\n",
" 70: 'send',\n",
" 71: 'want',\n",
" 72: \"i'll\",\n",
" 73: 'going',\n",
" 74: 'as',\n",
" 75: 'txt',\n",
" 76: 'one',\n",
" 77: 'about',\n",
" 78: 'lor',\n",
" 79: 'by',\n",
" 80: 'need',\n",
" 81: 'stop',\n",
" 82: 'r',\n",
" 83: 'home',\n",
" 84: 'still',\n",
" 85: 'today',\n",
" 86: 'sorry',\n",
" 87: 'n',\n",
" 88: 'our',\n",
" 89: 'see',\n",
" 90: 'back',\n",
" 91: 'reply',\n",
" 92: 'k',\n",
" 93: 'she',\n",
" 94: 'her',\n",
" 95: 'mobile',\n",
" 96: 'pls',\n",
" 97: 'later',\n",
" 98: 'da',\n",
" 99: 'any',\n",
" 100: 'been',\n",
" 101: 'tell',\n",
" 102: 'think',\n",
" 103: 'dont',\n",
" 104: \"don't\",\n",
" 105: 'week',\n",
" 106: 'please',\n",
" 107: 'take',\n",
" 108: 'new',\n",
" 109: 'they',\n",
" 110: 'hi',\n",
" 111: 'phone',\n",
" 112: 'did',\n",
" 113: 'some',\n",
" 114: 'ì',\n",
" 115: 'claim',\n",
" 116: '1',\n",
" 117: 'well',\n",
" 118: 'd',\n",
" 119: 'dear',\n",
" 120: 'has',\n",
" 121: 'great',\n",
" 122: 'here',\n",
" 123: 'an',\n",
" 124: 'much',\n",
" 125: 'night',\n",
" 126: 'who',\n",
" 127: 'oh',\n",
" 128: 'hope',\n",
" 129: 'him',\n",
" 130: 'msg',\n",
" 131: 'make',\n",
" 132: 'too',\n",
" 133: 'more',\n",
" 134: 'happy',\n",
" 135: 'wat',\n",
" 136: 'e',\n",
" 137: 'had',\n",
" 138: 'where',\n",
" 139: 'way',\n",
" 140: 'yes',\n",
" 141: 'work',\n",
" 142: 'give',\n",
" 143: 'hey',\n",
" 144: 'message',\n",
" 145: \"it's\",\n",
" 146: 'number',\n",
" 147: 'right',\n",
" 148: 'should',\n",
" 149: 'www',\n",
" 150: 'prize',\n",
" 151: 'tomorrow',\n",
" 152: 'cash',\n",
" 153: 'say',\n",
" 154: 'yeah',\n",
" 155: 'c',\n",
" 156: 'really',\n",
" 157: 'already',\n",
" 158: 'doing',\n",
" 159: 'why',\n",
" 160: 'said',\n",
" 161: 'after',\n",
" 162: 'them',\n",
" 163: 'ask',\n",
" 164: 'life',\n",
" 165: 'amp',\n",
" 166: '3',\n",
" 167: 'im',\n",
" 168: 'cos',\n",
" 169: 'thanks',\n",
" 170: 'meet',\n",
" 171: 'find',\n",
" 172: 'sent',\n",
" 173: 'last',\n",
" 174: 'would',\n",
" 175: 'anything',\n",
" 176: 'lol',\n",
" 177: 'morning',\n",
" 178: 'every',\n",
" 179: 'also',\n",
" 180: 'com',\n",
" 181: 'urgent',\n",
" 182: 'b',\n",
" 183: 'miss',\n",
" 184: 'babe',\n",
" 185: 'very',\n",
" 186: 'win',\n",
" 187: 'care',\n",
" 188: 'sure',\n",
" 189: 'contact',\n",
" 190: 'pick',\n",
" 191: 'keep',\n",
" 192: 'before',\n",
" 193: 't',\n",
" 194: 'let',\n",
" 195: 'buy',\n",
" 196: 'cant',\n",
" 197: 'nokia',\n",
" 198: 'won',\n",
" 199: 's',\n",
" 200: 'next',\n",
" 201: 'service',\n",
" 202: \"i've\",\n",
" 203: 'even',\n",
" 204: 'first',\n",
" 205: 'which',\n",
" 206: 'wait',\n",
" 207: '5',\n",
" 208: 'again',\n",
" 209: 'were',\n",
" 210: 'gonna',\n",
" 211: 'someone',\n",
" 212: 'thing',\n",
" 213: 'something',\n",
" 214: \"can't\",\n",
" 215: 'tone',\n",
" 216: 'customer',\n",
" 217: 'us',\n",
" 218: 'money',\n",
" 219: 'many',\n",
" 220: '50',\n",
" 221: 'min',\n",
" 222: '150p',\n",
" 223: 'sleep',\n",
" 224: 'nice',\n",
" 225: 'off',\n",
" 226: 'went',\n",
" 227: 'could',\n",
" 228: 'over',\n",
" 229: 'around',\n",
" 230: 'gud',\n",
" 231: 'friends',\n",
" 232: 'help',\n",
" 233: 'per',\n",
" 234: 'feel',\n",
" 235: 'uk',\n",
" 236: 'soon',\n",
" 237: 'haha',\n",
" 238: 'tonight',\n",
" 239: 'wish',\n",
" 240: 'place',\n",
" 241: 'hello',\n",
" 242: 'his',\n",
" 243: \"that's\",\n",
" 244: 'v',\n",
" 245: 'chat',\n",
" 246: 'guaranteed',\n",
" 247: '16',\n",
" 248: 'ya',\n",
" 249: 'told',\n",
" 250: 'special',\n",
" 251: 'other',\n",
" 252: 'sms',\n",
" 253: 'mins',\n",
" 254: 'wan',\n",
" 255: 'late',\n",
" 256: 'down',\n",
" 257: 'waiting',\n",
" 258: 'yet',\n",
" 259: 'name',\n",
" 260: 'getting',\n",
" 261: 'people',\n",
" 262: 'lunch',\n",
" 263: 'always',\n",
" 264: 'dun',\n",
" 265: 'best',\n",
" 266: 'ìï',\n",
" 267: 'fine',\n",
" 268: 'co',\n",
" 269: 'things',\n",
" 270: 'may',\n",
" 271: 'leave',\n",
" 272: 'thk',\n",
" 273: 'same',\n",
" 274: 'done',\n",
" 275: 'smile',\n",
" 276: 'class',\n",
" 277: 'never',\n",
" 278: 'try',\n",
" 279: 'yup',\n",
" 280: \"you're\",\n",
" 281: 'use',\n",
" 282: 'holiday',\n",
" 283: 'thought',\n",
" 284: 'draw',\n",
" 285: 'cs',\n",
" 286: 'friend',\n",
" 287: \"didn't\",\n",
" 288: 'y',\n",
" 289: 'having',\n",
" 290: 'few',\n",
" 291: 'year',\n",
" 292: 'finish',\n",
" 293: 'trying',\n",
" 294: '6',\n",
" 295: 'line',\n",
" 296: 'coming',\n",
" 297: '18',\n",
" 298: 'talk',\n",
" 299: 'meeting',\n",
" 300: 'being',\n",
" 301: 'job',\n",
" 302: '7',\n",
" 303: 'heart',\n",
" 304: 'world',\n",
" 305: 'house',\n",
" 306: 'person',\n",
" 307: 'stuff',\n",
" 308: 'cool',\n",
" 309: '1st',\n",
" 310: 'thats',\n",
" 311: 'days',\n",
" 312: 'better',\n",
" 313: 'å£1',\n",
" 314: 'mind',\n",
" 315: 'bit',\n",
" 316: 'eat',\n",
" 317: 'guess',\n",
" 318: 'because',\n",
" 319: 'receive',\n",
" 320: 'live',\n",
" 321: 'problem',\n",
" 322: 'shows',\n",
" 323: 'man',\n",
" 324: 'nothing',\n",
" 325: 'ready',\n",
" 326: 'chance',\n",
" 327: 'latest',\n",
" 328: 'god',\n",
" 329: 'yo',\n",
" 330: 'dat',\n",
" 331: 'month',\n",
" 332: 'word',\n",
" 333: 'liao',\n",
" 334: 'long',\n",
" 335: 'than',\n",
" 336: 'x',\n",
" 337: 'account',\n",
" 338: 'enjoy',\n",
" 339: 'wanna',\n",
" 340: 'car',\n",
" 341: 'bt',\n",
" 342: 'landline',\n",
" 343: 'cost',\n",
" 344: 'camera',\n",
" 345: 'check',\n",
" 346: 'lot',\n",
" 347: 'sir',\n",
" 348: 'jus',\n",
" 349: 'play',\n",
" 350: 'awarded',\n",
" 351: 'real',\n",
" 352: 'video',\n",
" 353: 'luv',\n",
" 354: 'dinner',\n",
" 355: 'half',\n",
" 356: 'birthday',\n",
" 357: 'å£1000',\n",
" 358: 'watching',\n",
" 359: 'might',\n",
" 360: 'offer',\n",
" 361: 'bed',\n",
" 362: '150ppm',\n",
" 363: 'into',\n",
" 364: 'shit',\n",
" 365: 'guys',\n",
" 366: 'orange',\n",
" 367: 'end',\n",
" 368: 'quite',\n",
" 369: 'pay',\n",
" 370: 'didnt',\n",
" 371: 'room',\n",
" 372: 'does',\n",
" 373: 'ringtone',\n",
" 374: 'code',\n",
" 375: 'dunno',\n",
" 376: 'girl',\n",
" 377: 'sweet',\n",
" 378: 'ill',\n",
" 379: 'called',\n",
" 380: 'aight',\n",
" 381: 'big',\n",
" 382: 'hear',\n",
" 383: 'weekend',\n",
" 384: 'boy',\n",
" 385: 'speak',\n",
" 386: 'ever',\n",
" 387: 'shall',\n",
" 388: 'reach',\n",
" 389: 'thanx',\n",
" 390: 'minutes',\n",
" 391: 'start',\n",
" 392: '10',\n",
" 393: 'shopping',\n",
" 394: 'ah',\n",
" 395: 'tv',\n",
" 396: 'little',\n",
" 397: 'box',\n",
" 398: 'network',\n",
" 399: 'asked',\n",
" 400: 'xxx',\n",
" 401: 'princess',\n",
" 402: 'actually',\n",
" 403: 'apply',\n",
" 404: 'between',\n",
" 405: 'bad',\n",
" 406: 'once',\n",
" 407: 'remember',\n",
" 408: 'another',\n",
" 409: 'face',\n",
" 410: 'watch',\n",
" 411: 'kiss',\n",
" 412: 'den',\n",
" 413: 'wk',\n",
" 414: 'office',\n",
" 415: 'baby',\n",
" 416: 'left',\n",
" 417: 'probably',\n",
" 418: 'wont',\n",
" 419: 'most',\n",
" 420: 'xmas',\n",
" 421: 'forgot',\n",
" 422: '9',\n",
" 423: 'fun',\n",
" 424: 'everything',\n",
" 425: 'early',\n",
" 426: 'lar',\n",
" 427: 'look',\n",
" 428: 'bring',\n",
" 429: 'dis',\n",
" 430: '2nd',\n",
" 431: 'made',\n",
" 432: 'able',\n",
" 433: 'put',\n",
" 434: 'po',\n",
" 435: 'rate',\n",
" 436: 'part',\n",
" 437: 'easy',\n",
" 438: 'll',\n",
" 439: 'bus',\n",
" 440: 'collect',\n",
" 441: 'sexy',\n",
" 442: 'came',\n",
" 443: 'sat',\n",
" 444: 'missing',\n",
" 445: 'says',\n",
" 446: 'mob',\n",
" 447: 'enough',\n",
" 448: 'pa',\n",
" 449: 'de',\n",
" 450: 'selected',\n",
" 451: 'plan',\n",
" 452: 'tmr',\n",
" 453: 'calls',\n",
" 454: 'looking',\n",
" 455: 'town',\n",
" 456: 'award',\n",
" 457: 'since',\n",
" 458: 'while',\n",
" 459: 'anyway',\n",
" 460: 'price',\n",
" 461: \"he's\",\n",
" 462: 'fuck',\n",
" 463: 'havent',\n",
" 464: 'leh',\n",
" 465: 'dad',\n",
" 466: 'until',\n",
" 467: '8',\n",
" 468: '000',\n",
" 469: 'messages',\n",
" 470: 'mail',\n",
" 471: 'must',\n",
" 472: 'times',\n",
" 473: 'plz',\n",
" 474: 'school',\n",
" 475: 'those',\n",
" 476: 'double',\n",
" 477: 'maybe',\n",
" 478: 'plus',\n",
" 479: 'update',\n",
" 480: 'true',\n",
" 481: 'thank',\n",
" 482: '500',\n",
" 483: 'lei',\n",
" 484: 'wife',\n",
" 485: 'away',\n",
" 486: 'hour',\n",
" 487: 'collection',\n",
" 488: 'wanted',\n",
" 489: 'though',\n",
" 490: 'without',\n",
" 491: 'two',\n",
" 492: 'means',\n",
" 493: 'important',\n",
" 494: 'pain',\n",
" 495: 'å£100',\n",
" 496: 'gift',\n",
" 497: 'saw',\n",
" 498: '10p',\n",
" 499: 'yours',\n",
" 500: 'hair',\n",
" 501: 'texts',\n",
" 502: 'hav',\n",
" 503: 'wen',\n",
" 504: 'g',\n",
" 505: 'm',\n",
" 506: 'wake',\n",
" 507: 'alright',\n",
" 508: 'details',\n",
" 509: 'goes',\n",
" 510: 'valid',\n",
" 511: 'stay',\n",
" 512: 'wot',\n",
" 513: 'game',\n",
" 514: 'yesterday',\n",
" 515: 'music',\n",
" 516: 'working',\n",
" 517: 'haf',\n",
" 518: 'colour',\n",
" 519: \"''\",\n",
" 520: 'okay',\n",
" 521: 'join',\n",
" 522: 'lose',\n",
" 523: 'bored',\n",
" 524: 'net',\n",
" 525: 'food',\n",
" 526: 're',\n",
" 527: 'top',\n",
" 528: 'coz',\n",
" 529: 'chikku',\n",
" 530: \"we're\",\n",
" 531: 'entry',\n",
" 532: 'wid',\n",
" 533: 'hours',\n",
" 534: 'either',\n",
" 535: 'friendship',\n",
" 536: 'wif',\n",
" 537: 'tried',\n",
" 538: \"haven't\",\n",
" 539: \"how's\",\n",
" 540: 'oso',\n",
" 541: 'book',\n",
" 542: 'question',\n",
" 543: 'show',\n",
" 544: 'set',\n",
" 545: 'test',\n",
" 546: 'shop',\n",
" 547: 'id',\n",
" 548: 'feeling',\n",
" 549: 'å£2000',\n",
" 550: 'decimal',\n",
" 551: \"there's\",\n",
" 552: 'goin',\n",
" 553: 'afternoon',\n",
" 554: 'family',\n",
" 555: 'weekly',\n",
" 556: 'private',\n",
" 557: 'points',\n",
" 558: 'till',\n",
" 559: 'attempt',\n",
" 560: 'online',\n",
" 561: 'driving',\n",
" 562: 'vouchers',\n",
" 563: 'order',\n",
" 564: 'run',\n",
" 565: 'hurt',\n",
" 566: 'together',\n",
" 567: 'leaving',\n",
" 568: 'mean',\n",
" 569: 'unsubscribe',\n",
" 570: 'charge',\n",
" 571: 'delivery',\n",
" 572: 'email',\n",
" 573: '750',\n",
" 574: 'juz',\n",
" 575: 'makes',\n",
" 576: \"we'll\",\n",
" 577: 'congrats',\n",
" 578: 'answer',\n",
" 579: 'http',\n",
" 580: 'dude',\n",
" 581: 'aft',\n",
" 582: 'evening',\n",
" 583: 'started',\n",
" 584: 'ard',\n",
" 585: 'ring',\n",
" 586: 'todays',\n",
" 587: 'trip',\n",
" 588: 'guy',\n",
" 589: 'til',\n",
" 590: 'worry',\n",
" 591: 'missed',\n",
" 592: 'movie',\n",
" 593: 'beautiful',\n",
" 594: 'wants',\n",
" 595: 'å£500',\n",
" 596: 'making',\n",
" 597: 'sae',\n",
" 598: 'sleeping',\n",
" 599: 'awesome',\n",
" 600: 'walk',\n",
" 601: 'post',\n",
" 602: 'land',\n",
" 603: 'full',\n",
" 604: 'happen',\n",
" 605: 'services',\n",
" 606: 'pounds',\n",
" 607: 'smiling',\n",
" 608: 'change',\n",
" 609: 'national',\n",
" 610: 'pub',\n",
" 611: 'close',\n",
" 612: 'comes',\n",
" 613: 'neva',\n",
" 614: 'saying',\n",
" 615: 'calling',\n",
" 616: 'gd',\n",
" 617: 'taking',\n",
" 618: 'nite',\n",
" 619: 'else',\n",
" 620: 'tot',\n",
" 621: 'both',\n",
" 622: 'company',\n",
" 623: \"won't\",\n",
" 624: 'thinking',\n",
" 625: 'huh',\n",
" 626: 'forget',\n",
" 627: 'statement',\n",
" 628: 'expires',\n",
" 629: 'words',\n",
" 630: \"what's\",\n",
" 631: 'sad',\n",
" 632: '11',\n",
" 633: 'sch',\n",
" 634: 'sounds',\n",
" 635: 'abt',\n",
" 636: 'await',\n",
" 637: 'drive',\n",
" 638: 'hot',\n",
" 639: 'yourself',\n",
" 640: 'sister',\n",
" 641: 'old',\n",
" 642: 'available',\n",
" 643: 'break',\n",
" 644: 'college',\n",
" 645: 'ha',\n",
" 646: 'tones',\n",
" 647: 'believe',\n",
" 648: 'knw',\n",
" 649: 'noe',\n",
" 650: 'choose',\n",
" 651: 'mobileupd8',\n",
" 652: 'msgs',\n",
" 653: 'story',\n",
" 654: 'these',\n",
" 655: 'date',\n",
" 656: 'bonus',\n",
" 657: 'final',\n",
" 658: 'eve',\n",
" 659: 'identifier',\n",
" 660: 'tho',\n",
" 661: 'read',\n",
" 662: 'pic',\n",
" 663: 'second',\n",
" 664: 'opt',\n",
" 665: 'angry',\n",
" 666: 'smoke',\n",
" 667: 'search',\n",
" 668: '12hrs',\n",
" 669: 'brother',\n",
" 670: 'address',\n",
" 671: 'years',\n",
" 672: 'gr8',\n",
" 673: 'pics',\n",
" 674: 'tomo',\n",
" 675: 'alone',\n",
" 676: 'type',\n",
" 677: 'used',\n",
" 678: 'drop',\n",
" 679: 'wrong',\n",
" 680: 'okie',\n",
" 681: 'don',\n",
" 682: 'games',\n",
" 683: 'å£5000',\n",
" 684: 'crazy',\n",
" 685: 'o',\n",
" 686: '08000930705',\n",
" 687: \"she's\",\n",
" 688: 'dating',\n",
" 689: \"c's\",\n",
" 690: 'cause',\n",
" 691: 'xx',\n",
" 692: 'head',\n",
" 693: 'kind',\n",
" 694: 'wit',\n",
" 695: 'takes',\n",
" 696: 'rite',\n",
" 697: 'finished',\n",
" 698: '100',\n",
" 699: 'whatever',\n",
" 700: 'club',\n",
" 701: 'loving',\n",
" 702: 'wil',\n",
" 703: 'decided',\n",
" 704: 'found',\n",
" 705: '\\x89û',\n",
" 706: 'voucher',\n",
" 707: 'anytime',\n",
" 708: 'mum',\n",
" 709: 'lets',\n",
" 710: 'pm',\n",
" 711: 'minute',\n",
" 712: 'winner',\n",
" 713: 'tel',\n",
" 714: '08000839402',\n",
" 715: '30',\n",
" 716: 'dreams',\n",
" 717: 'anyone',\n",
" 718: 'drink',\n",
" 719: 'bank',\n",
" 720: 'camcorder',\n",
" 721: 'hmm',\n",
" 722: 'smth',\n",
" 723: 'prob',\n",
" 724: 'mu',\n",
" 725: 'touch',\n",
" 726: 'sis',\n",
" 727: 'took',\n",
" 728: \"'\",\n",
" 729: 'mate',\n",
" 730: 'whats',\n",
" 731: 'wkly',\n",
" 732: 'worth',\n",
" 733: 'savamob',\n",
" 734: 'å£3',\n",
" 735: 'content',\n",
" 736: 'chennai',\n",
" 737: 'lucky',\n",
" 738: 'busy',\n",
" 739: 'fancy',\n",
" 740: 'tc',\n",
" 741: 'hit',\n",
" 742: 'mine',\n",
" 743: 'telling',\n",
" 744: 'poly',\n",
" 745: 'nope',\n",
" 746: 'sun',\n",
" 747: 'earlier',\n",
" 748: 'fucking',\n",
" 749: 'lots',\n",
" 750: 'news',\n",
" 751: 'hows',\n",
" 752: 'snow',\n",
" 753: 'ni8',\n",
" 754: 'whole',\n",
" 755: 'case',\n",
" 756: 'operator',\n",
" 757: 'cut',\n",
" 758: 'finally',\n",
" 759: 'nt',\n",
" 760: 'light',\n",
" 761: 'blue',\n",
" 762: 'mrng',\n",
" 763: 'happiness',\n",
" 764: 'secret',\n",
" 765: 'thinks',\n",
" 766: 'quiz',\n",
" 767: 'f',\n",
" 768: 'mates',\n",
" 769: 'motorola',\n",
" 770: 'far',\n",
" 771: 'sea',\n",
" 772: \"doesn't\",\n",
" 773: 'phones',\n",
" 774: 'girls',\n",
" 775: 'credit',\n",
" 776: 'card',\n",
" 777: 'rs',\n",
" 778: '86688',\n",
" 779: 'darlin',\n",
" 780: 'w',\n",
" 781: 'boytoy',\n",
" 782: 'yr',\n",
" 783: 'everyone',\n",
" 784: 'info',\n",
" 785: '2003',\n",
" 786: '800',\n",
" 787: 'open',\n",
" 788: 'lesson',\n",
" 789: 'oredi',\n",
" 790: 'weeks',\n",
" 791: 'pobox',\n",
" 792: 'treat',\n",
" 793: 'offers',\n",
" 794: 'hard',\n",
" 795: 'visit',\n",
" 796: 'frm',\n",
" 797: 'happened',\n",
" 798: 'hungry',\n",
" 799: 'fast',\n",
" 800: 'felt',\n",
" 801: 'row',\n",
" 802: 'couple',\n",
" 803: 'wonderful',\n",
" 804: 'saturday',\n",
" 805: 'cum',\n",
" 806: 'park',\n",
" 807: 'least',\n",
" 808: 'hold',\n",
" 809: '8007',\n",
" 810: 'listen',\n",
" 811: 'mah',\n",
" 812: 'friday',\n",
" 813: 'bout',\n",
" 814: 'lovely',\n",
" 815: 'valued',\n",
" 816: 'b4',\n",
" 817: 'their',\n",
" 818: 'stupid',\n",
" 819: 'through',\n",
" 820: 'enter',\n",
" 821: 'frnds',\n",
" 822: 'currently',\n",
" 823: 'gas',\n",
" 824: 'almost',\n",
" 825: 'txts',\n",
" 826: 'fone',\n",
" 827: \"you'll\",\n",
" 828: 'reading',\n",
" 829: 'reason',\n",
" 830: 'dnt',\n",
" 831: 'log',\n",
" 832: 'gotta',\n",
" 833: 'via',\n",
" 834: 'india',\n",
" 835: 'auction',\n",
" 836: 'rental',\n",
" 837: 'parents',\n",
" 838: 'download',\n",
" 839: 'each',\n",
" 840: 'mom',\n",
" 841: 'complimentary',\n",
" 842: 'congratulations',\n",
" 843: 'uncle',\n",
" 844: 'un',\n",
" 845: 'redeemed',\n",
" 846: '04',\n",
" 847: 'die',\n",
" 848: 'hee',\n",
" 849: 'call2optout',\n",
" 850: 'needs',\n",
" 851: '87066',\n",
" 852: 'frnd',\n",
" 853: 'meant',\n",
" 854: 'askd',\n",
" 855: 'ends',\n",
" 856: 'goodmorning',\n",
" 857: 'player',\n",
" 858: 'sunday',\n",
" 859: 'charged',\n",
" 860: 'age',\n",
" 861: 'welcome',\n",
" 862: 'wonder',\n",
" 863: 'freemsg',\n",
" 864: 'wow',\n",
" 865: 'gn',\n",
" 866: \"wasn't\",\n",
" 867: 'etc',\n",
" 868: 'monday',\n",
" 869: 'bslvyl',\n",
" 870: '20',\n",
" 871: 'extra',\n",
" 872: 'mobiles',\n",
" 873: 'within',\n",
" 874: 'song',\n",
" 875: 'yar',\n",
" 876: 'talking',\n",
" 877: '0800',\n",
" 878: 'outside',\n",
" 879: 'picking',\n",
" 880: 'tired',\n",
" 881: 'entered',\n",
" 882: 'point',\n",
" 883: 'heard',\n",
" 884: 'reveal',\n",
" 885: 'comp',\n",
" 886: 'simple',\n",
" 887: 'swing',\n",
" 888: 'press',\n",
" 889: 'carlos',\n",
" 890: 'project',\n",
" 891: 'luck',\n",
" 892: 'mayb',\n",
" 893: 'checking',\n",
" 894: 'seeing',\n",
" 895: 'balance',\n",
" 896: 'unlimited',\n",
" 897: 'sort',\n",
" 898: 'paper',\n",
" 899: 'txting',\n",
" 900: 'mr',\n",
" 901: \"i'd\",\n",
" 902: 'discount',\n",
" 903: 'pretty',\n",
" 904: \"you've\",\n",
" 905: 'surprise',\n",
" 906: 'information',\n",
" 907: 'grins',\n",
" 908: 'support',\n",
" 909: 'red',\n",
" 910: 'gone',\n",
" 911: 'invited',\n",
" 912: 'figure',\n",
" 913: 'jay',\n",
" 914: 'clean',\n",
" 915: 'bcoz',\n",
" 916: 'kids',\n",
" 917: 'abiola',\n",
" 918: 'cheap',\n",
" 919: 'orchard',\n",
" 920: 'booked',\n",
" 921: 'na',\n",
" 922: 'oops',\n",
" 923: 'sending',\n",
" 924: 'nah',\n",
" 925: 'å£350',\n",
" 926: 'lovable',\n",
" 927: 'correct',\n",
" 928: 'ans',\n",
" 929: 'nyt',\n",
" 930: 'leaves',\n",
" 931: 'å£250',\n",
" 932: 'std',\n",
" 933: 'sell',\n",
" 934: 'gets',\n",
" 935: '12',\n",
" 936: 'st',\n",
" 937: '00',\n",
" 938: '03',\n",
" 939: 'fact',\n",
" 940: 'direct',\n",
" 941: 'spend',\n",
" 942: 'christmas',\n",
" 943: 'sex',\n",
" 944: 'pass',\n",
" 945: 'confirm',\n",
" 946: 'gym',\n",
" 947: \"b'day\",\n",
" 948: 'questions',\n",
" 949: 'blood',\n",
" 950: 'ass',\n",
" 951: 'callertune',\n",
" 952: 'th',\n",
" 953: 'understand',\n",
" 954: 'ago',\n",
" 955: 'numbers',\n",
" 956: 'sub',\n",
" 957: '2nite',\n",
" 958: 'months',\n",
" 959: 'tonite',\n",
" 960: 'reward',\n",
" 961: 'fr',\n",
" 962: 'glad',\n",
" 963: 'darren',\n",
" 964: 'father',\n",
" 965: 'å£10',\n",
" 966: 'terms',\n",
" 967: 'p',\n",
" 968: 'fri',\n",
" 969: 'loved',\n",
" 970: 'john',\n",
" 971: 'ten',\n",
" 972: 'truth',\n",
" 973: 'dogging',\n",
" 974: 'convey',\n",
" 975: 'exam',\n",
" 976: 'march',\n",
" 977: 'suite342',\n",
" 978: '2lands',\n",
" 979: 'feels',\n",
" 980: 'gal',\n",
" 981: 'drugs',\n",
            " 982: 'laugh',\n",
            " 983: 'dream',\n",
            " 984: 'accept',\n",
            " 985: 'credits',\n",
            " 986: 'comuk',\n",
            " 987: 'caller',\n",
            " 988: 'computer',\n",
            " 989: 'valentines',\n",
            " 990: 'link',\n",
            " 991: 'poor',\n",
            " 992: 'buying',\n",
            " 993: 'specially',\n",
            " 994: 'semester',\n",
            " 995: 'side',\n",
            " 996: 'worries',\n",
            " 997: 'admirer',\n",
            " 998: 'ex',\n",
            " 999: 'match',\n",
            " 1000: 'ringtones',\n",
            " ...}"
          ]
        },
        "metadata": {},
        "execution_count": 10
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "**Creating model and Adding the layers**"
    ],
    "metadata": {
      "id": "7q7hlyuTjKFV"
    }
  },
  {
    "cell_type": "code",
    "source": [
      "TOT_SIZE = len(tok.word_index)+1\n",
      "lstm_model = Sequential()\n",
      "lstm_model.add(Embedding(TOT_SIZE, 32, input_length=max_len))\n",
      "lstm_model.add(LSTM(100))\n",
      "lstm_model.add(Dropout(0.4))\n",
      "lstm_model.add(Dense(20, activation=\"relu\"))\n",
      "lstm_model.add(Dropout(0.3))\n",
        "lstm_model.add(Dense(1, activation = \"sigmoid\"))"
      ],
      "metadata": {
        "id": "j3RtA4Ihe_9G"
      },
      "execution_count": 11,
      "outputs": []
    },
    {
      "cell_type": "markdown",
      "source": [
        "**Compile the model**"
      ],
      "metadata": {
        "id": "pK177WZ7jB1d"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "lstm_model.compile(loss = \"binary_crossentropy\", optimizer = \"adam\",
metrics = [\"accuracy\"])\n",
        "lstm_model.summary()"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "gItoFBABi9QG",
        "outputId": "c5224612-474c-49d4-ba79-be868c882ff7"
      },
      "execution_count": 12,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Model: \"sequential\"\n",
            "_________________________________________________________________\n",
            " Layer (type)                Output Shape              Param #   \n",
            "=================================================================\n",
            " embedding (Embedding)       (None, 150, 32)           262528    \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: 317,769\n",
            "Trainable params: 317,769\n",
            "Non-trainable params: 0\n",
            "_________________________________________________________________\n"
          ]
        }
      ]
    },
    {
      "cell_type": "markdown",
      "source": [
        "**Fit the model**"
      ],
      "metadata": {
        "id": "8qR_Ov4cib1w"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "lstm_model.fit(sequences_matrix,Y_train,batch_size=128,epochs=10,\n",
        "          validation_split=0.2,\n",
        "          workers=10,\n",
        "
callbacks=[EarlyStopping(monitor='val_loss',min_delta=0.0001)])"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "I_gluT2Qfjer",
        "outputId": "c251fb56-445d-4ac7-f0e4-62bc03967380"
      },
      "execution_count": 13,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
     "text": [
      "Epoch 1/10\n",
      "30/30 [==============================] - 17s 442ms/step - loss: 0.4943
- accuracy: 0.8522 - val_loss: 0.3745 - val_accuracy: 0.8586\n",
      "Epoch 2/10\n",
      "30/30 [==============================] - 18s 599ms/step - loss: 0.2779
- accuracy: 0.8968 - val_loss: 0.1414 - val_accuracy: 0.9652\n"
     ]
    },
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "<keras.callbacks.History at 0x7f9ee6bf6190>"
      ]
     },
     "metadata": {},
     "execution_count": 13
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Save the model**"
   ],
   "metadata": {
    "id": "t_Rr6SLciOvg"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "lstm_model.save('sms.h5')"
   ],
   "metadata": {
    "id": "uDoxxzfuh8Z4"
   },
   "execution_count": 14,
   "outputs": []
  },
  {
   "cell_type": "markdown",
   "source": [
    "**Test the model**"
   ],
      "metadata": {
        "id": "zva5H2EhlAdZ"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "test_sequences = tok.texts_to_sequences(X_test)\n",
        "test_sequences_matrix = pad_sequences(test_sequences,maxlen=max_len)"
      ],
      "metadata": {
        "id": "V2QFA689hTL7"
      },
      "execution_count": 15,
      "outputs": []
    },
    {
      "cell_type": "code",
      "source": [
        "acc = lstm_model.evaluate(test_sequences_matrix,Y_test)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "QnWrbEWvhhKy",
        "outputId": "f2a1dc5a-7608-4660-a73b-3bd4efae7be2"
      },
      "execution_count": 16,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "27/27 [==============================] - 1s 36ms/step - loss: 0.1619 -
accuracy: 0.9569\n"
          ]
        }
      ]
    },
    {
      "cell_type": "code",
      "source": [
        "print('Test set\\n  Loss: {:0.3f}\\n  Accuracy:
{:0.3f}'.format(acc[0],acc[1]))"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "RGQsqrjXhuMa",
        "outputId": "9237259a-bd9f-4dee-8d4d-216913f1214d"
      },
      "execution_count": 17,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "Test set\n",
            "  Loss: 0.162\n",
            "  Accuracy: 0.957\n"
          ]
        }
      ]
    }
  ]
}
```