{"metadata":{"kernelspec":{"language":"python","display\_name":"Python 3","name":"python3"},"language\_info":{"name":"python","version":"3.7.10","mimetype":"text/x-python","codemirror\_mode":{"name":"ipython","version":3},"pygments\_lexer":"ipython3","nbconvert\_exporter":"python","file\_extension":".py"}},"nbformat\_minor":4,"nbformat":4,"cells":[{"cell\_type":"markdown","source":"# \*\*\*\*Spam Message Classification using LSTM\*\*\*\*","metadata":{}},{"cell\_type":"markdown","source":"## 1.Import the Necessary Libraries","metadata":{}},{"cell\_type":"code","source":"import numpy as np\nimport pandas as pd\nimport os\nimport seaborn as sns\nimport matplotlib.pyplot as plt\n%matplotlib inline\nfor dirname, \_, filenames in os.walk('/kaggle/input'):\n for filename in filenames:\n print(os.path.join(dirname, filename))\n","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:02.239872Z","iopub.execute\_input":"2022-10-24T19:10:02.240492Z","iopub.status.idle":"2022-10-24T19:10:03.096453Z","shell.execute\_reply.started":"2022-10-24T19:10:02.240435Z","shell.execute\_reply":"2022-10-24T19:10:03.095272Z"},"trusted":true},"execution\_count":2,"outputs":[{"name":"stdout","text":"/kaggle/input/sms-spam-collection-dataset/spam.csv\n","output\_type":"stream"}]},{"cell\_type":"markdown","source":"\*\*\*\*2. Reading the .csv dataset\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"data=pd.read\_csv(\"../input/sms-spam-collection-dataset/spam.csv\",encoding=\"latin\")\ndata.head()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.098112Z","iopub.execute\_input":"2022-10-24T19:10:03.098765Z","iopub.status.idle":"2022-10-24T19:10:03.176425Z","shell.execute\_reply.started":"2022-10-24T19:10:03.098714Z","shell.execute\_reply":"2022-10-24T19:10:03.175300Z"},"trusted":true},"execution\_count":3,"outputs":[{"execution\_count":3,"output\_type":"execute\_result","data":{"text/plain":" v1 v2 Unnamed: 2 \\\n0 ham Go until jurong point, crazy.. Available only ... NaN \n1 ham Ok lar... Joking wif u oni... NaN \n2 spam Free entry in 2 a wkly comp to win FA Cup fina... NaN \n3 ham U dun say so early hor... U c already then say... NaN \n4 ham Nah I don't think he goes to usf, he lives aro... NaN \n\n Unnamed: 3 Unnamed: 4 \n0 NaN NaN \n1 NaN NaN \n2 NaN NaN \n3 NaN NaN \n4 NaN NaN ","text/html":"<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>"},"metadata":{}}]},{"cell\_type":"code","source":"data.columns","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.178761Z","iopub.execute\_input":"2022-10-24T19:10:03.179533Z","iopub.status.idle":"2022-10-24T19:10:03.187473Z","shell.execute\_reply.started":"2022-10-24T19:10:03.179482Z","shell.execute\_reply":"2022-10-24T19:10:03.186504Z"},"trusted":true},"execution\_count":4,"outputs":[{"execution\_count":4,"output\_type":"execute\_result","data":{"text/plain":"Index(['v1', 'v2', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')"},"metadata":{}}]},{"cell\_type":"markdown","source":"\*\*\*\*3. Drop the unnamed Columns\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"data=data.drop(columns=[\"Unnamed: 2\",\"Unnamed: 3\",\"Unnamed: 4\"])","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.189783Z","iopub.execute\_input":"2022-10-24T19:10:03.190917Z","iopub.status.idle":"2022-10-24T19:10:03.200280Z","shell.execute\_reply.started":"2022-10-24T19:10:03.190612Z","shell.execute\_reply":"2022-10-24T19:10:03.199069Z"},"trusted":true},"execution\_count":5,"outputs":[]},{"cell\_type":"markdown","source":"\*\*\*\*4. Renaming Column names sensible\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"data=data.rename(\n{\n \"v1\":\"Category\",\n \"v2\":\"Message\"\n},\n axis=1\n)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.201564Z","iopub.execute\_input":"2022-10-24T19:10:03.201941Z","iopub.status.idle":"2022-10-24T19:10:03.214540Z","shell.execute\_reply.started":"2022-10-24T19:10:03.201904Z","shell.execute\_reply":"2022-10-24T19:10:03.213334Z"},"trusted":true},"execution\_count":6,"outputs":[]},{"cell\_type":"code","source":"data.head()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.218668Z","iopub.execute\_input":"2022-10-24T19:10:03.219672Z","iopub.status.idle":"2022-10-24T19:10:03.233081Z","shell.execute\_reply.started":"2022-10-24T19:10:03.219616Z","shell.execute\_reply":"2022-10-24T19:10:03.231569Z"},"trusted":true},"execution\_count":7,"outputs":[{"execution\_count":7,"output\_type":"execute\_result","data":{"text/plain":" Category Message\n0 ham Go until jurong point, crazy.. Available only ...\n1 ham Ok lar... Joking wif u oni...\n2 spam Free entry in 2 a wkly comp to win FA Cup fina...\n3 ham U dun say so early hor... U c already then say...\n4 ham Nah I don't think he goes to usf, he lives aro...","text/html":"<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>Category</th>\n <th>Message</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 </tr>\n <tr>\n <th>1</th>\n <td>ham</td>\n <td>Ok lar... Joking wif u oni...</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 </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 </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 </tr>\n </tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell\_type":"markdown","source":"\*\*\*\*5. Check for null values in dataset\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"data.isnull().sum()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.235631Z","iopub.execute\_input":"2022-10-24T19:10:03.236080Z","iopub.status.idle":"2022-10-24T19:10:03.249179Z","shell.execute\_reply.started":"2022-10-24T19:10:03.236033Z","shell.execute\_reply":"2022-10-24T19:10:03.248321Z"},"trusted":true},"execution\_count":8,"outputs":[{"execution\_count":8,"output\_type":"execute\_result","data":{"text/plain":"Category 0\nMessage 0\ndtype: int64"},"metadata":{}}]},{"cell\_type":"code","source":"data.info()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.252620Z","iopub.execute\_input":"2022-10-24T19:10:03.253279Z","iopub.status.idle":"2022-10-24T19:10:03.292918Z","shell.execute\_reply.started":"2022-10-24T19:10:03.253229Z","shell.execute\_reply":"2022-10-24T19:10:03.291720Z"},"trusted":true},"execution\_count":9,"outputs":[{"name":"stdout","text":"<class 'pandas.core.frame.DataFrame'>\nRangeIndex: 5572 entries, 0 to 5571\nData columns (total 2 columns):\n # Column Non-Null Count Dtype \n--- ------ -------------- ----- \n 0 Category 5572 non-null object\n 1 Message 5572 non-null object\ndtypes: object(2)\nmemory usage: 87.2+ KB\n","output\_type":"stream"}]},{"cell\_type":"markdown","source":"\*\*\*\*6.Creating a new Field to store the Message Lengths\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"data[\"Message Length\"]=data[\"Message\"].apply(len)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.295879Z","iopub.execute\_input":"2022-10-24T19:10:03.296745Z","iopub.status.idle":"2022-10-24T19:10:03.306522Z","shell.execute\_reply.started":"2022-10-24T19:10:03.296691Z","shell.execute\_reply":"2022-10-24T19:10:03.305269Z"},"trusted":true},"execution\_count":10,"outputs":[]},{"cell\_type":"markdown","source":"\*\*\*\*7. Histogram Inference of Message Lengths of Spam and Non-spam messages\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"fig=plt.figure(figsize=(12,8))\nsns.histplot(\n x=data[\"Message Length\"],\n hue=data[\"Category\"]\n)\nplt.title(\"ham & spam messege length comparision\")\nplt.show()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:03.308175Z","iopub.execute\_input":"2022-10-24T19:10:03.308832Z","iopub.status.idle":"2022-10-24T19:10:04.060142Z","shell.execute\_reply.started":"2022-10-24T19:10:03.308780Z","shell.execute\_reply":"2022-10-24T19:10:04.059065Z"},"trusted":true},"execution\_count":11,"outputs":[{"output\_type":"display\_data","data":{"text/plain":"<Figure size 864x576 with 1 Axes>","image/png":"\n"},"metadata":{"needs\_background":"light"}}]},{"cell\_type":"code","source":"ham\_desc=data[data[\"Category\"]==\"ham\"][\"Message Length\"].describe()\nspam\_desc=data[data[\"Category\"]==\"spam\"][\"Message Length\"].describe()\n\nprint(\"Ham Messege Length Description:\\n\",ham\_desc)\nprint(\"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\")\nprint(\"Spam Message Length Description:\\n\",spam\_desc)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.061689Z","iopub.execute\_input":"2022-10-24T19:10:04.062304Z","iopub.status.idle":"2022-10-24T19:10:04.083748Z","shell.execute\_reply.started":"2022-10-24T19:10:04.062255Z","shell.execute\_reply":"2022-10-24T19:10:04.082850Z"},"trusted":true},"execution\_count":12,"outputs":[{"name":"stdout","text":"Ham Messege Length Description:\n count 4825.000000\nmean 71.023627\nstd 58.016023\nmin 2.000000\n25% 33.000000\n50% 52.000000\n75% 92.000000\nmax 910.000000\nName: Message Length, dtype: float64\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nSpam Message Length Description:\n count 747.000000\nmean 138.866131\nstd 29.183082\nmin 13.000000\n25% 132.500000\n50% 149.000000\n75% 157.000000\nmax 224.000000\nName: Message Length, dtype: float64\n","output\_type":"stream"}]},{"cell\_type":"code","source":"data.describe(include=\"all\")","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.085185Z","iopub.execute\_input":"2022-10-24T19:10:04.085701Z","iopub.status.idle":"2022-10-24T19:10:04.116444Z","shell.execute\_reply.started":"2022-10-24T19:10:04.085661Z","shell.execute\_reply":"2022-10-24T19:10:04.115239Z"},"trusted":true},"execution\_count":13,"outputs":[{"execution\_count":13,"output\_type":"execute\_result","data":{"text/plain":" Category Message Message Length\ncount 5572 5572 5572.000000\nunique 2 5169 NaN\ntop ham Sorry, I'll call later NaN\nfreq 4825 30 NaN\nmean NaN NaN 80.118808\nstd NaN NaN 59.690841\nmin NaN NaN 2.000000\n25% NaN NaN 36.000000\n50% NaN NaN 61.000000\n75% NaN NaN 121.000000\nmax NaN NaN 910.000000","text/html":"<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>Category</th>\n <th>Message</th>\n <th>Message Length</th>\n </tr>\n </thead>\n <tbody>\n <tr>\n <th>count</th>\n <td>5572</td>\n <td>5572</td>\n <td>5572.000000</td>\n </tr>\n <tr>\n <th>unique</th>\n <td>2</td>\n <td>5169</td>\n <td>NaN</td>\n </tr>\n <tr>\n <th>top</th>\n <td>ham</td>\n <td>Sorry, I'll call later</td>\n <td>NaN</td>\n </tr>\n <tr>\n <th>freq</th>\n <td>4825</td>\n <td>30</td>\n <td>NaN</td>\n </tr>\n <tr>\n <th>mean</th>\n <td>NaN</td>\n <td>NaN</td>\n <td>80.118808</td>\n </tr>\n <tr>\n <th>std</th>\n <td>NaN</td>\n <td>NaN</td>\n <td>59.690841</td>\n </tr>\n <tr>\n <th>min</th>\n <td>NaN</td>\n <td>NaN</td>\n <td>2.000000</td>\n </tr>\n <tr>\n <th>25%</th>\n <td>NaN</td>\n <td>NaN</td>\n <td>36.000000</td>\n </tr>\n <tr>\n <th>50%</th>\n <td>NaN</td>\n <td>NaN</td>\n <td>61.000000</td>\n </tr>\n <tr>\n <th>75%</th>\n <td>NaN</td>\n <td>NaN</td>\n <td>121.000000</td>\n </tr>\n <tr>\n <th>max</th>\n <td>NaN</td>\n <td>NaN</td>\n <td>910.000000</td>\n </tr>\n </tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell\_type":"markdown","source":"\*\*\*\*8. Visualizing count of messages of Spam and Non Spam\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"data[\"Category\"].value\_counts()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.118065Z","iopub.execute\_input":"2022-10-24T19:10:04.118721Z","iopub.status.idle":"2022-10-24T19:10:04.130611Z","shell.execute\_reply.started":"2022-10-24T19:10:04.118669Z","shell.execute\_reply":"2022-10-24T19:10:04.129145Z"},"trusted":true},"execution\_count":14,"outputs":[{"execution\_count":14,"output\_type":"execute\_result","data":{"text/plain":"ham 4825\nspam 747\nName: Category, dtype: int64"},"metadata":{}}]},{"cell\_type":"code","source":"sns.countplot(\n data=data,\n x=\"Category\"\n)\nplt.title(\"ham vs spam\")\nplt.show()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.133436Z","iopub.execute\_input":"2022-10-24T19:10:04.134267Z","iopub.status.idle":"2022-10-24T19:10:04.279114Z","shell.execute\_reply.started":"2022-10-24T19:10:04.134214Z","shell.execute\_reply":"2022-10-24T19:10:04.277884Z"},"trusted":true},"execution\_count":15,"outputs":[{"output\_type":"display\_data","data":{"text/plain":"<Figure size 432x288 with 1 Axes>","image/png":"\n"},"metadata":{"needs\_background":"light"}}]},{"cell\_type":"code","source":"ham\_count=data[\"Category\"].value\_counts()[0]\nspam\_count=data[\"Category\"].value\_counts()[1]\n\ntotal\_count=data.shape[0]\n\nprint(\"Ham contains:{:.2f}% of total data.\".format(ham\_count/total\_count\*100))\nprint(\"Spam contains:{:.2f}% of total data.\".format(spam\_count/total\_count\*100))","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.280745Z","iopub.execute\_input":"2022-10-24T19:10:04.281391Z","iopub.status.idle":"2022-10-24T19:10:04.299266Z","shell.execute\_reply.started":"2022-10-24T19:10:04.281337Z","shell.execute\_reply":"2022-10-24T19:10:04.297746Z"},"trusted":true},"execution\_count":16,"outputs":[{"name":"stdout","text":"Ham contains:86.59% of total data.\nSpam contains:13.41% of total data.\n","output\_type":"stream"}]},{"cell\_type":"markdown","source":"\*\*\*\*9. Undersampling to Genralize Model and Balance Spam and Ham quantities in dataset\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"minority\_len=len(data[data[\"Category\"]==\"spam\"])\nmajority\_len=len(data[data[\"Category\"]==\"ham\"])\nminority\_indices=data[data[\"Category\"]==\"spam\"].index\nmajority\_indices=data[data[\"Category\"]==\"ham\"].index\nrandom\_majority\_indices=np.random.choice(\n majority\_indices,\n size=minority\_len,\n replace=False\n)\n\nundersampled\_indices=np.concatenate([minority\_indices,random\_majority\_indices])\ndf=data.loc[undersampled\_indices]\ndf=df.sample(frac=1)\ndf=df.reset\_index()\ndf=df.drop(\n columns=[\"index\"],\n)\n","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.301158Z","iopub.execute\_input":"2022-10-24T19:10:04.302281Z","iopub.status.idle":"2022-10-24T19:10:04.321490Z","shell.execute\_reply.started":"2022-10-24T19:10:04.302228Z","shell.execute\_reply":"2022-10-24T19:10:04.320190Z"},"trusted":true},"execution\_count":17,"outputs":[]},{"cell\_type":"code","source":"df.shape","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.325469Z","iopub.execute\_input":"2022-10-24T19:10:04.326524Z","iopub.status.idle":"2022-10-24T19:10:04.335046Z","shell.execute\_reply.started":"2022-10-24T19:10:04.326458Z","shell.execute\_reply":"2022-10-24T19:10:04.333177Z"},"trusted":true},"execution\_count":18,"outputs":[{"execution\_count":18,"output\_type":"execute\_result","data":{"text/plain":"(1494, 3)"},"metadata":{}}]},{"cell\_type":"code","source":"df[\"Category\"].value\_counts()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.336861Z","iopub.execute\_input":"2022-10-24T19:10:04.337416Z","iopub.status.idle":"2022-10-24T19:10:04.349938Z","shell.execute\_reply.started":"2022-10-24T19:10:04.337364Z","shell.execute\_reply":"2022-10-24T19:10:04.348682Z"},"trusted":true},"execution\_count":19,"outputs":[{"execution\_count":19,"output\_type":"execute\_result","data":{"text/plain":"ham 747\nspam 747\nName: Category, dtype: int64"},"metadata":{}}]},{"cell\_type":"code","source":"sns.countplot(\n data=df,\n x=\"Category\"\n)\nplt.title(\"ham vs spam\")\nplt.show()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.351795Z","iopub.execute\_input":"2022-10-24T19:10:04.352407Z","iopub.status.idle":"2022-10-24T19:10:04.517748Z","shell.execute\_reply.started":"2022-10-24T19:10:04.352361Z","shell.execute\_reply":"2022-10-24T19:10:04.516618Z"},"trusted":true},"execution\_count":20,"outputs":[{"output\_type":"display\_data","data":{"text/plain":"<Figure size 432x288 with 1 Axes>","image/png":"\n"},"metadata":{"needs\_background":"light"}}]},{"cell\_type":"markdown","source":"Display the head of new \*\*df\*\*","metadata":{}},{"cell\_type":"code","source":"df.head()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.519296Z","iopub.execute\_input":"2022-10-24T19:10:04.519904Z","iopub.status.idle":"2022-10-24T19:10:04.533858Z","shell.execute\_reply.started":"2022-10-24T19:10:04.519856Z","shell.execute\_reply":"2022-10-24T19:10:04.532971Z"},"trusted":true},"execution\_count":21,"outputs":[{"execution\_count":21,"output\_type":"execute\_result","data":{"text/plain":" Category Message Message Length\n0 spam FREE>Ringtone! Reply REAL or POLY eg REAL1 1. ... 158\n1 spam URGENT! We are trying to contact U Todays draw... 157\n2 ham Ok ill send you with in &lt;DECIMAL&gt; ok. 45\n3 ham Oh just getting even with u.... u? 34\n4 spam A link to your picture has been sent. You can ... 96","text/html":"<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>Category</th>\n <th>Message</th>\n <th>Message Length</th>\n </tr>\n </thead>\n <tbody>\n <tr>\n <th>0</th>\n <td>spam</td>\n <td>FREE&gt;Ringtone! Reply REAL or POLY eg REAL1 1. ...</td>\n <td>158</td>\n </tr>\n <tr>\n <th>1</th>\n <td>spam</td>\n <td>URGENT! We are trying to contact U Todays draw...</td>\n <td>157</td>\n </tr>\n <tr>\n <th>2</th>\n <td>ham</td>\n <td>Ok ill send you with in &amp;lt;DECIMAL&amp;gt; ok.</td>\n <td>45</td>\n </tr>\n <tr>\n <th>3</th>\n <td>ham</td>\n <td>Oh just getting even with u.... u?</td>\n <td>34</td>\n </tr>\n <tr>\n <th>4</th>\n <td>spam</td>\n <td>A link to your picture has been sent. You can ...</td>\n <td>96</td>\n </tr>\n </tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell\_type":"markdown","source":"\*\*\*\*10. Binary Encoding of Spam and Ham Categories\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"df[\"Label\"]=df[\"Category\"].map(\n {\n \"ham\":0,\n \"spam\":1\n }\n)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.535464Z","iopub.execute\_input":"2022-10-24T19:10:04.535903Z","iopub.status.idle":"2022-10-24T19:10:04.547404Z","shell.execute\_reply.started":"2022-10-24T19:10:04.535855Z","shell.execute\_reply":"2022-10-24T19:10:04.546160Z"},"trusted":true},"execution\_count":22,"outputs":[]},{"cell\_type":"code","source":"df.head()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.549181Z","iopub.execute\_input":"2022-10-24T19:10:04.549942Z","iopub.status.idle":"2022-10-24T19:10:04.569516Z","shell.execute\_reply.started":"2022-10-24T19:10:04.549885Z","shell.execute\_reply":"2022-10-24T19:10:04.567227Z"},"trusted":true},"execution\_count":23,"outputs":[{"execution\_count":23,"output\_type":"execute\_result","data":{"text/plain":" Category Message Message Length \\\n0 spam FREE>Ringtone! Reply REAL or POLY eg REAL1 1. ... 158 \n1 spam URGENT! We are trying to contact U Todays draw... 157 \n2 ham Ok ill send you with in &lt;DECIMAL&gt; ok. 45 \n3 ham Oh just getting even with u.... u? 34 \n4 spam A link to your picture has been sent. You can ... 96 \n\n Label \n0 1 \n1 1 \n2 0 \n3 0 \n4 1 ","text/html":"<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>Category</th>\n <th>Message</th>\n <th>Message Length</th>\n <th>Label</th>\n </tr>\n </thead>\n <tbody>\n <tr>\n <th>0</th>\n <td>spam</td>\n <td>FREE&gt;Ringtone! Reply REAL or POLY eg REAL1 1. ...</td>\n <td>158</td>\n <td>1</td>\n </tr>\n <tr>\n <th>1</th>\n <td>spam</td>\n <td>URGENT! We are trying to contact U Todays draw...</td>\n <td>157</td>\n <td>1</td>\n </tr>\n <tr>\n <th>2</th>\n <td>ham</td>\n <td>Ok ill send you with in &amp;lt;DECIMAL&amp;gt; ok.</td>\n <td>45</td>\n <td>0</td>\n </tr>\n <tr>\n <th>3</th>\n <td>ham</td>\n <td>Oh just getting even with u.... u?</td>\n <td>34</td>\n <td>0</td>\n </tr>\n <tr>\n <th>4</th>\n <td>spam</td>\n <td>A link to your picture has been sent. You can ...</td>\n <td>96</td>\n <td>1</td>\n </tr>\n </tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell\_type":"markdown","source":"\*\*\*\*11. Import Necessary Libraries to perform Word Tokenization\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"import re\nimport nltk\nfrom nltk.corpus import stopwords\nfrom nltk.stem import PorterStemmer\n\nstemmer=PorterStemmer()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:04.591608Z","iopub.execute\_input":"2022-10-24T19:10:04.592013Z","iopub.status.idle":"2022-10-24T19:10:05.468505Z","shell.execute\_reply.started":"2022-10-24T19:10:04.591972Z","shell.execute\_reply":"2022-10-24T19:10:05.467421Z"},"trusted":true},"execution\_count":24,"outputs":[]},{"cell\_type":"code","source":"corpus=[]\nfor message in df[\"Message\"]:\n message=re.sub(\"[^a-zA-Z]\",\" \",message)\n message=message.lower()\n message=message.split()\n message=[stemmer.stem(words)\n for words in message\n if words not in set(stopwords.words(\"english\"))\n ]\n message=\" \".join(message)\n corpus.append(message)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:05.471181Z","iopub.execute\_input":"2022-10-24T19:10:05.471629Z","iopub.status.idle":"2022-10-24T19:10:10.529923Z","shell.execute\_reply.started":"2022-10-24T19:10:05.471582Z","shell.execute\_reply":"2022-10-24T19:10:10.529054Z"},"trusted":true},"execution\_count":25,"outputs":[]},{"cell\_type":"markdown","source":"\*\*\*\*12. Perform One Hot on Corpus\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"from tensorflow.keras.preprocessing.text import one\_hot\nvocab\_size=10000\n\noneHot\_doc=[one\_hot(words,n=vocab\_size)\n for words in corpus\n ]","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:10.531300Z","iopub.execute\_input":"2022-10-24T19:10:10.531883Z","iopub.status.idle":"2022-10-24T19:10:16.532983Z","shell.execute\_reply.started":"2022-10-24T19:10:10.531836Z","shell.execute\_reply":"2022-10-24T19:10:16.531832Z"},"trusted":true},"execution\_count":26,"outputs":[]},{"cell\_type":"code","source":"df[\"Message Length\"].describe()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.535185Z","iopub.execute\_input":"2022-10-24T19:10:16.535794Z","iopub.status.idle":"2022-10-24T19:10:16.552102Z","shell.execute\_reply.started":"2022-10-24T19:10:16.535745Z","shell.execute\_reply":"2022-10-24T19:10:16.550839Z"},"trusted":true},"execution\_count":27,"outputs":[{"execution\_count":27,"output\_type":"execute\_result","data":{"text/plain":"count 1494.000000\nmean 104.491299\nstd 60.362332\nmin 2.000000\n25% 49.000000\n50% 114.000000\n75% 153.000000\nmax 910.000000\nName: Message Length, dtype: float64"},"metadata":{}}]},{"cell\_type":"code","source":"fig=plt.figure(figsize=(12,8))\nsns.kdeplot(\n x=df[\"Message Length\"],\n hue=df[\"Category\"]\n)\nplt.title(\"ham & spam messege length comparision\")\nplt.show()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.553602Z","iopub.execute\_input":"2022-10-24T19:10:16.554125Z","iopub.status.idle":"2022-10-24T19:10:16.832915Z","shell.execute\_reply.started":"2022-10-24T19:10:16.554067Z","shell.execute\_reply":"2022-10-24T19:10:16.832067Z"},"trusted":true},"execution\_count":28,"outputs":[{"output\_type":"display\_data","data":{"text/plain":"<Figure size 864x576 with 1 Axes>","image/png":"\n"},"metadata":{"needs\_background":"light"}}]},{"cell\_type":"code","source":"from tensorflow.keras.preprocessing.sequence import pad\_sequences\nsentence\_len=200\nembedded\_doc=pad\_sequences(\n oneHot\_doc,\n maxlen=sentence\_len,\n padding=\"pre\"\n)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.834627Z","iopub.execute\_input":"2022-10-24T19:10:16.835387Z","iopub.status.idle":"2022-10-24T19:10:16.850505Z","shell.execute\_reply.started":"2022-10-24T19:10:16.835340Z","shell.execute\_reply":"2022-10-24T19:10:16.849664Z"},"trusted":true},"execution\_count":29,"outputs":[]},{"cell\_type":"code","source":"extract\_features=pd.DataFrame(\n data=embedded\_doc\n)\ntarget=df[\"Label\"]","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.852065Z","iopub.execute\_input":"2022-10-24T19:10:16.852771Z","iopub.status.idle":"2022-10-24T19:10:16.858781Z","shell.execute\_reply.started":"2022-10-24T19:10:16.852725Z","shell.execute\_reply":"2022-10-24T19:10:16.858029Z"},"trusted":true},"execution\_count":30,"outputs":[]},{"cell\_type":"code","source":"df\_final=pd.concat([extract\_features,target],axis=1)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.860452Z","iopub.execute\_input":"2022-10-24T19:10:16.861330Z","iopub.status.idle":"2022-10-24T19:10:16.873596Z","shell.execute\_reply.started":"2022-10-24T19:10:16.861282Z","shell.execute\_reply":"2022-10-24T19:10:16.872357Z"},"trusted":true},"execution\_count":31,"outputs":[]},{"cell\_type":"code","source":"df\_final.head()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.887891Z","iopub.execute\_input":"2022-10-24T19:10:16.888544Z","iopub.status.idle":"2022-10-24T19:10:16.915667Z","shell.execute\_reply.started":"2022-10-24T19:10:16.888482Z","shell.execute\_reply":"2022-10-24T19:10:16.914443Z"},"trusted":true},"execution\_count":32,"outputs":[{"execution\_count":32,"output\_type":"execute\_result","data":{"text/plain":" 0 1 2 3 4 5 6 7 8 9 ... 191 192 193 194 195 196 \\\n0 0 0 0 0 0 0 0 0 0 0 ... 8116 8983 7883 1884 5957 5877 \n1 0 0 0 0 0 0 0 0 0 0 ... 9989 7682 5710 5519 2447 1240 \n2 0 0 0 0 0 0 0 0 0 0 ... 0 0 3310 6099 7761 9276 \n3 0 0 0 0 0 0 0 0 0 0 ... 0 0 0 0 8194 7945 \n4 0 0 0 0 0 0 0 0 0 0 ... 5677 7440 8481 9975 2366 4841 \n\n 197 198 199 Label \n0 266 1527 5846 1 \n1 3994 6950 3655 1 \n2 4679 2205 3310 0 \n3 3841 266 266 0 \n4 4320 4320 4672 1 \n\n[5 rows x 201 columns]","text/html":"<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>0</th>\n <th>1</th>\n <th>2</th>\n <th>3</th>\n <th>4</th>\n <th>5</th>\n <th>6</th>\n <th>7</th>\n <th>8</th>\n <th>9</th>\n <th>...</th>\n <th>191</th>\n <th>192</th>\n <th>193</th>\n <th>194</th>\n <th>195</th>\n <th>196</th>\n <th>197</th>\n <th>198</th>\n <th>199</th>\n <th>Label</th>\n </tr>\n </thead>\n <tbody>\n <tr>\n <th>0</th>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>...</td>\n <td>8116</td>\n <td>8983</td>\n <td>7883</td>\n <td>1884</td>\n <td>5957</td>\n <td>5877</td>\n <td>266</td>\n <td>1527</td>\n <td>5846</td>\n <td>1</td>\n </tr>\n <tr>\n <th>1</th>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>...</td>\n <td>9989</td>\n <td>7682</td>\n <td>5710</td>\n <td>5519</td>\n <td>2447</td>\n <td>1240</td>\n <td>3994</td>\n <td>6950</td>\n <td>3655</td>\n <td>1</td>\n </tr>\n <tr>\n <th>2</th>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>...</td>\n <td>0</td>\n <td>0</td>\n <td>3310</td>\n <td>6099</td>\n <td>7761</td>\n <td>9276</td>\n <td>4679</td>\n <td>2205</td>\n <td>3310</td>\n <td>0</td>\n </tr>\n <tr>\n <th>3</th>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>...</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>8194</td>\n <td>7945</td>\n <td>3841</td>\n <td>266</td>\n <td>266</td>\n <td>0</td>\n </tr>\n <tr>\n <th>4</th>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>0</td>\n <td>...</td>\n <td>5677</td>\n <td>7440</td>\n <td>8481</td>\n <td>9975</td>\n <td>2366</td>\n <td>4841</td>\n <td>4320</td>\n <td>4320</td>\n <td>4672</td>\n <td>1</td>\n </tr>\n </tbody>\n</table>\n<p>5 rows × 201 columns</p>\n</div>"},"metadata":{}}]},{"cell\_type":"markdown","source":"\*\*13. Splitting Dependent and Independent Variables\*\*","metadata":{}},{"cell\_type":"code","source":"X=df\_final.drop(\"Label\",axis=1)\ny=df\_final[\"Label\"]","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.917484Z","iopub.execute\_input":"2022-10-24T19:10:16.917975Z","iopub.status.idle":"2022-10-24T19:10:16.925694Z","shell.execute\_reply.started":"2022-10-24T19:10:16.917926Z","shell.execute\_reply":"2022-10-24T19:10:16.924444Z"},"trusted":true},"execution\_count":33,"outputs":[]},{"cell\_type":"markdown","source":"\*\*\*\*14. Train, test and Validation Split\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"from sklearn.model\_selection import train\_test\_split","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.927716Z","iopub.execute\_input":"2022-10-24T19:10:16.928220Z","iopub.status.idle":"2022-10-24T19:10:16.936425Z","shell.execute\_reply.started":"2022-10-24T19:10:16.928172Z","shell.execute\_reply":"2022-10-24T19:10:16.935448Z"},"trusted":true},"execution\_count":34,"outputs":[]},{"cell\_type":"code","source":"X\_trainval,X\_test,y\_trainval,y\_test=train\_test\_split(\n X,\n y,\n random\_state=42,\n test\_size=0.15\n)\n","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.937992Z","iopub.execute\_input":"2022-10-24T19:10:16.938747Z","iopub.status.idle":"2022-10-24T19:10:16.955745Z","shell.execute\_reply.started":"2022-10-24T19:10:16.938696Z","shell.execute\_reply":"2022-10-24T19:10:16.954517Z"},"trusted":true},"execution\_count":35,"outputs":[]},{"cell\_type":"code","source":"X\_train,X\_val,y\_train,y\_val=train\_test\_split(\n X\_trainval,\n y\_trainval,\n random\_state=42,\n test\_size=0.15\n)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.958421Z","iopub.execute\_input":"2022-10-24T19:10:16.958783Z","iopub.status.idle":"2022-10-24T19:10:16.991967Z","shell.execute\_reply.started":"2022-10-24T19:10:16.958746Z","shell.execute\_reply":"2022-10-24T19:10:16.990522Z"},"trusted":true},"execution\_count":36,"outputs":[]},{"cell\_type":"markdown","source":"\*\*\*\*15.Building a Sequential Model\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"from tensorflow.keras.layers import LSTM\nfrom tensorflow.keras.layers import Dense\nfrom tensorflow.keras.layers import Embedding\nfrom tensorflow.keras.models import Sequential\n","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:16.996041Z","iopub.execute\_input":"2022-10-24T19:10:16.996635Z","iopub.status.idle":"2022-10-24T19:10:17.002522Z","shell.execute\_reply.started":"2022-10-24T19:10:16.996574Z","shell.execute\_reply":"2022-10-24T19:10:17.001396Z"},"trusted":true},"execution\_count":37,"outputs":[]},{"cell\_type":"code","source":"model=Sequential()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:17.004323Z","iopub.execute\_input":"2022-10-24T19:10:17.004959Z","iopub.status.idle":"2022-10-24T19:10:17.046634Z","shell.execute\_reply.started":"2022-10-24T19:10:17.004910Z","shell.execute\_reply":"2022-10-24T19:10:17.045551Z"},"trusted":true},"execution\_count":38,"outputs":[]},{"cell\_type":"code","source":"feature\_num=100\nmodel.add(\n Embedding(\n input\_dim=vocab\_size,\n output\_dim=feature\_num,\n input\_length=sentence\_len\n )\n)\nmodel.add(\n LSTM(\n units=128\n )\n)\n\nmodel.add(\n Dense(\n units=1,\n activation=\"sigmoid\"\n )\n)\nmodel.summary()","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:17.049491Z","iopub.execute\_input":"2022-10-24T19:10:17.049905Z","iopub.status.idle":"2022-10-24T19:10:17.434593Z","shell.execute\_reply.started":"2022-10-24T19:10:17.049860Z","shell.execute\_reply":"2022-10-24T19:10:17.433603Z"},"trusted":true},"execution\_count":39,"outputs":[{"name":"stdout","text":"Model: \"sequential\"\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\nLayer (type) Output Shape Param # \n=================================================================\nembedding (Embedding) (None, 200, 100) 1000000 \n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\nlstm (LSTM) (None, 128) 117248 \n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\ndense (Dense) (None, 1) 129 \n=================================================================\nTotal params: 1,117,377\nTrainable params: 1,117,377\nNon-trainable params: 0\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n","output\_type":"stream"}]},{"cell\_type":"code","source":"from tensorflow.keras.optimizers import Adam\nmodel.compile(\n optimizer=Adam(\n learning\_rate=0.001\n ),\n loss=\"binary\_crossentropy\",\n metrics=[\"accuracy\"]\n)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:17.438169Z","iopub.execute\_input":"2022-10-24T19:10:17.438485Z","iopub.status.idle":"2022-10-24T19:10:17.456008Z","shell.execute\_reply.started":"2022-10-24T19:10:17.438452Z","shell.execute\_reply":"2022-10-24T19:10:17.454160Z"},"trusted":true},"execution\_count":40,"outputs":[]},{"cell\_type":"markdown","source":"\*\*\*\*16. Model Fitting\*\*\*\*","metadata":{}},{"cell\_type":"code","source":"history=model.fit(\n X\_train,\n y\_train,\n validation\_data=(\n X\_val,\n y\_val\n ),\n epochs=10\n)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:10:17.458198Z","iopub.execute\_input":"2022-10-24T19:10:17.459013Z","iopub.status.idle":"2022-10-24T19:13:49.021304Z","shell.execute\_reply.started":"2022-10-24T19:10:17.458961Z","shell.execute\_reply":"2022-10-24T19:13:49.020067Z"},"trusted":true},"execution\_count":41,"outputs":[{"name":"stdout","text":"Epoch 1/10\n34/34 [==============================] - 24s 633ms/step - loss: 0.6324 - accuracy: 0.6331 - val\_loss: 0.4218 - val\_accuracy: 0.8377\nEpoch 2/10\n34/34 [==============================] - 21s 608ms/step - loss: 0.3045 - accuracy: 0.9257 - val\_loss: 0.1631 - val\_accuracy: 0.9529\nEpoch 3/10\n34/34 [==============================] - 21s 609ms/step - loss: 0.1046 - accuracy: 0.9689 - val\_loss: 0.1231 - val\_accuracy: 0.9581\nEpoch 4/10\n34/34 [==============================] - 21s 621ms/step - loss: 0.0465 - accuracy: 0.9880 - val\_loss: 0.1293 - val\_accuracy: 0.9581\nEpoch 5/10\n34/34 [==============================] - 21s 613ms/step - loss: 0.0342 - accuracy: 0.9895 - val\_loss: 0.1252 - val\_accuracy: 0.9686\nEpoch 6/10\n34/34 [==============================] - 21s 615ms/step - loss: 0.0179 - accuracy: 0.9951 - val\_loss: 0.1366 - val\_accuracy: 0.9686\nEpoch 7/10\n34/34 [==============================] - 21s 614ms/step - loss: 0.0121 - accuracy: 0.9968 - val\_loss: 0.1314 - val\_accuracy: 0.9634\nEpoch 8/10\n34/34 [==============================] - 21s 619ms/step - loss: 0.0222 - accuracy: 0.9944 - val\_loss: 0.1479 - val\_accuracy: 0.9634\nEpoch 9/10\n34/34 [==============================] - 21s 614ms/step - loss: 0.0077 - accuracy: 0.9989 - val\_loss: 0.1624 - val\_accuracy: 0.9634\nEpoch 10/10\n34/34 [==============================] - 21s 614ms/step - loss: 0.0077 - accuracy: 0.9976 - val\_loss: 0.1751 - val\_accuracy: 0.9634\n","output\_type":"stream"}]},{"cell\_type":"code","source":"metrics = pd.DataFrame(history.history)\nmetrics.rename(columns = {'loss': 'Training\_Loss', 'accuracy': 'Training\_Accuracy', 'val\_loss': 'Validation\_Loss', 'val\_accuracy': 'Validation\_Accuracy'}, inplace = True)\ndef plot\_graph\_acc(var1, var2, string):\n metrics[[var1, var2]].plot()\n plt.title('Training and Validation ' + string)\n plt.xlabel ('Number of epochs')\n plt.ylabel(string)\n plt.legend([var1, var2])","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:13:49.023674Z","iopub.execute\_input":"2022-10-24T19:13:49.024023Z","iopub.status.idle":"2022-10-24T19:13:49.035310Z","shell.execute\_reply.started":"2022-10-24T19:13:49.023986Z","shell.execute\_reply":"2022-10-24T19:13:49.034334Z"},"trusted":true},"execution\_count":42,"outputs":[]},{"cell\_type":"code","source":"plot\_graph\_acc('Training\_Accuracy', 'Validation\_Accuracy', 'accuracy')","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:13:49.036836Z","iopub.execute\_input":"2022-10-24T19:13:49.037532Z","iopub.status.idle":"2022-10-24T19:13:49.261844Z","shell.execute\_reply.started":"2022-10-24T19:13:49.037483Z","shell.execute\_reply":"2022-10-24T19:13:49.260734Z"},"trusted":true},"execution\_count":43,"outputs":[{"output\_type":"display\_data","data":{"text/plain":"<Figure size 432x288 with 1 Axes>","image/png":"\n"},"metadata":{"needs\_background":"light"}}]},{"cell\_type":"code","source":"y\_pred=model.predict(X\_test)\ny\_pred=(y\_pred>0.5)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:13:49.263521Z","iopub.execute\_input":"2022-10-24T19:13:49.264176Z","iopub.status.idle":"2022-10-24T19:13:50.795719Z","shell.execute\_reply.started":"2022-10-24T19:13:49.264125Z","shell.execute\_reply":"2022-10-24T19:13:50.794739Z"},"trusted":true},"execution\_count":44,"outputs":[]},{"cell\_type":"code","source":"model.save('Spam\_SMS\_classifier.h5')","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:13:50.797185Z","iopub.execute\_input":"2022-10-24T19:13:50.797967Z","iopub.status.idle":"2022-10-24T19:13:50.856554Z","shell.execute\_reply.started":"2022-10-24T19:13:50.797917Z","shell.execute\_reply":"2022-10-24T19:13:50.855473Z"},"trusted":true},"execution\_count":45,"outputs":[]},{"cell\_type":"markdown","source":"\*\*17. Evaluating the Model\*\*","metadata":{}},{"cell\_type":"code","source":"from sklearn.metrics import accuracy\_score,confusion\_matrix","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:13:50.857979Z","iopub.execute\_input":"2022-10-24T19:13:50.858447Z","iopub.status.idle":"2022-10-24T19:13:50.863609Z","shell.execute\_reply.started":"2022-10-24T19:13:50.858404Z","shell.execute\_reply":"2022-10-24T19:13:50.862543Z"},"trusted":true},"execution\_count":46,"outputs":[]},{"cell\_type":"code","source":"score=accuracy\_score(y\_test,y\_pred)\nprint(\"Test Score:{:.2f}%\".format(score\*100))","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:13:50.865169Z","iopub.execute\_input":"2022-10-24T19:13:50.865778Z","iopub.status.idle":"2022-10-24T19:13:50.891954Z","shell.execute\_reply.started":"2022-10-24T19:13:50.865735Z","shell.execute\_reply":"2022-10-24T19:13:50.890586Z"},"trusted":true},"execution\_count":47,"outputs":[{"name":"stdout","text":"Test Score:96.89%\n","output\_type":"stream"}]},{"cell\_type":"code","source":"cm=confusion\_matrix(y\_test,y\_pred)\nfig=plt.figure(figsize=(12,8))\nsns.heatmap(\n cm,\n annot=True,\n)\nplt.title(\"Confusion Matrix\")\ncm","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:13:50.893821Z","iopub.execute\_input":"2022-10-24T19:13:50.894590Z","iopub.status.idle":"2022-10-24T19:13:51.204505Z","shell.execute\_reply.started":"2022-10-24T19:13:50.894537Z","shell.execute\_reply":"2022-10-24T19:13:51.203608Z"},"trusted":true},"execution\_count":48,"outputs":[{"execution\_count":48,"output\_type":"execute\_result","data":{"text/plain":"array([[104, 0],\n [ 7, 114]])"},"metadata":{}},{"output\_type":"display\_data","data":{"text/plain":"<Figure size 864x576 with 2 Axes>","image/png":"\n"},"metadata":{"needs\_background":"light"}}]},{"cell\_type":"markdown","source":"\*\*18. Function to Test the Model on a Random message\*\*","metadata":{}},{"cell\_type":"code","source":"def classify\_message(model,message):\n for sentences in message:\n sentences=nltk.sent\_tokenize(message)\n for sentence in sentences:\n words=re.sub(\"[^a-zA-Z]\",\" \",sentence)\n if words not in set(stopwords.words('english')):\n word=nltk.word\_tokenize(words)\n word=\" \".join(word) \n oneHot=[one\_hot(word,n=vocab\_size)]\n text=pad\_sequences(oneHot,maxlen=sentence\_len,padding=\"pre\")\n predict=model.predict(text)\n if predict>0.5:\n print(\"It is a spam\")\n print(\"predict score: \", predict[0][0])\n else:\n print(\"It is not a spam\")\n print(\"predict score: \", predict[0][0])\n ","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:19:21.347784Z","iopub.execute\_input":"2022-10-24T19:19:21.348350Z","iopub.status.idle":"2022-10-24T19:19:21.356492Z","shell.execute\_reply.started":"2022-10-24T19:19:21.348306Z","shell.execute\_reply":"2022-10-24T19:19:21.355309Z"},"trusted":true},"execution\_count":62,"outputs":[]},{"cell\_type":"code","source":"message1=\"I am having my Tests right now. Will call back as soon as possible! Till then be safe wherever you are. Be Alert of any hazard\"\nmessage2=\"Your Rs.8850 welcome bonus is ready to be credited. Download Junglee Rummy now. Claim Bonus on your first deposit prize pool\"\n","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:23:53.722985Z","iopub.execute\_input":"2022-10-24T19:23:53.723431Z","iopub.status.idle":"2022-10-24T19:23:53.728032Z","shell.execute\_reply.started":"2022-10-24T19:23:53.723389Z","shell.execute\_reply":"2022-10-24T19:23:53.726763Z"},"trusted":true},"execution\_count":80,"outputs":[]},{"cell\_type":"code","source":"classify\_message(model,message1)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:23:56.825850Z","iopub.execute\_input":"2022-10-24T19:23:56.826747Z","iopub.status.idle":"2022-10-24T19:23:57.062242Z","shell.execute\_reply.started":"2022-10-24T19:23:56.826692Z","shell.execute\_reply":"2022-10-24T19:23:57.061103Z"},"trusted":true},"execution\_count":81,"outputs":[{"name":"stdout","text":"It is not a spam\npredict score: 0.037389785\n","output\_type":"stream"}]},{"cell\_type":"code","source":"classify\_message(model,message2)","metadata":{"execution":{"iopub.status.busy":"2022-10-24T19:24:00.121976Z","iopub.execute\_input":"2022-10-24T19:24:00.122590Z","iopub.status.idle":"2022-10-24T19:24:00.314110Z","shell.execute\_reply.started":"2022-10-24T19:24:00.122536Z","shell.execute\_reply":"2022-10-24T19:24:00.312823Z"},"trusted":true},"execution\_count":82,"outputs":[{"name":"stdout","text":"It is a spam\npredict score: 0.9936712\n","output\_type":"stream"}]}]}