

```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "fVBYM8gzAjQT"
      },
      "outputs": [],
      "source": [
        "import pandas as pd\n",
        "import numpy as np\n",
```

```
"from sklearn.model_selection import train_test_split\n",  
"from sklearn.feature_extraction.text import TfidfVectorizer\n",  
"from sklearn.linear_model import LogisticRegression\n",  
"from sklearn.metrics import accuracy_score"  
]  
,  
{  
  "cell_type": "code",  
  "source": [  
    "df = pd.read_csv(\"/content/mail_data.csv\")\n",  
    "df.head()"  
  ],  
  "metadata": {  
    "colab": {  
      "base_uri": "https://localhost:8080/",  
      "height": 206  
    },  
    "id": "ulcjQAW8BXS7",  
    "outputId": "2148b2fd-35f8-454d-ef31-d7ef13192517"  
  },  
  "execution_count": null,  
  "outputs": [  
    {  
      "output_type": "execute_result",  
      "data": {  
        "text/plain": [  

```

```
" Category                Message\n",
"0  ham  Go until jurong point, crazy.. Available only ...\n",
"1  ham           Ok lar... Joking wif u oni...\n",
"2  spam  Free entry in 2 a wkly comp to win FA Cup fina...\n",
"3  ham  U dun say so early hor... U c already then say...\n",
"4  ham  Nah I don't think he goes to usf, he lives aro..."
```

```
],
```

```
"text/html": [
```

```
"\n",
```

```
"\n",
```

```
" <div id=\"df-760f8252-4e3f-44f0-8515-6af852939ed4\">\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",\n" <thead>\n",\n" <tr style=\"text-align: right;\">\n",\n" <th></th>\n",\n" <th>Category</th>\n",\n" <th>Message</th>\n",\n" </tr>\n",\n" </thead>\n",\n" <tbody>\n",\n" <tr>\n",\n" <th>0</th>\n",\n" <td>ham</td>\n",\n" <td>Go until jurong point, crazy.. Available only ...</td>\n",\n" </tr>\n",\n" <tr>\n",\n" <th>1</th>\n",\n" <td>ham</td>\n",\n" <td>Ok lar... Joking wif u oni...</td>\n",\n" </tr>\n",\n" <tr>\n",\n" <th>2</th>\n",\n" <td>spam</td>\n",\n" <td>Free entry in 2 a wkly comp to win FA Cup fina...</td>\n",\n" </tr>\n",\n" <tr>\n",\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>\n",

"  <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-760f8252-4e3f-44f0-8515-6af852939ed4')\" \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 2.06-2.06.94zm-11 11L8.5 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-.78.78-2.05 0 2.83L4 21.41c.39.39.95 1.41.59 1.41 0 1.02-.2 1.41-.59 1.41-7.78-7.78-2.81-2.81-.78-.78-2.07 0-2.86zM5.41 20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
"  </svg>\n",
" </button>\n",

```

```
"\n",
"\n",
"\n",
"  <div id=\"df-225d466f-b484-4126-a379-78e2338621b0\">\n",
"    <button class=\"colab-df-quickchart\" onclick=\"quickchart('df-225d466f-b484-4126-a379-78e2338621b0')\">\n",
"      title=\"Suggest charts.\"\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",
"    <g>\n",
"      <path d=\"M19 3H5c-1.1 0-.9 2 2v14c0 1.1.9 2 2 2h14c1.1 0 2-.9 2-2V5c0-1.1-.9-2-2-2z\"M9 17H7v-7h2v7zm4 0h-2V7h2v10zm4 0h-2v-4h2v4z\"/>\n",
"    </g>\n",
"</svg>\n",
"  </button>\n",
"</div>\n",
"\n",
"<style>\n",
"  .colab-df-quickchart {\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-quickchart: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-quickchart {\n",
" background-color: #3B4455;\n",
" fill: #D2E3FC;\n",
" }\n",
"\n",
" [theme=dark] .colab-df-quickchart: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",
"   async function quickchart(key) {\n",
"     const containerElement = document.querySelector('#' + key);\n",
"     const charts = await google.colab.kernel.invokeFunction(\n",
"       'suggestCharts', [key], {});\n",
"   }\n",
" </script>\n",
"\n",
" <script>\n",
"\n",
"function displayQuickchartButton(domScope) {\n",
"  let quickchartButtonEl =\n",
"    domScope.querySelector('#df-225d466f-b484-4126-a379-78e2338621b0  

button.colab-df-quickchart');\n",
"  quickchartButtonEl.style.display =\n",
"    google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"}\n",
"\n",
"  displayQuickchartButton(document);\n",
" </script>\n",
" <style>\n",
" .colab-df-container {\n",
"   display: flex;\n",
"   flex-wrap: wrap;\n",
"   gap: 12px;\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-760f8252-4e3f-44f0-8515-6af852939ed4\n",
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-760f8252-4e3f-44f0-8515-\n",
6af852939ed4');\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\"\n",
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table\n",
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": 41
}
]
},
{
  "cell_type": "code",
  "source": [
    "df.info()"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },

```

```

    "id": "2Tc5hWLvBXVZ",
    "outputId": "31fef96a-1202-4793-dc70-51bcc5735d86"
  },
  "execution_count": null,
  "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",
        "---  -
        0   Category  5572 non-null   object\n",
        "1   Message   5572 non-null   object\n",
        "dtypes: object(2)\n",
        "memory usage: 87.2+ KB\n"
      ]
    }
  ],
  "cell_type": "code",
  "source": [
    "df.isnull().sum()"
  ]

```

```
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "f1SjRQAjgliO",
  "outputId": "0044d2f7-040d-4cf9-948e-0b7ba079e4b2"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "Category  0\n",
        "Message  0\n",
        "dtype: int64"
      ]
    },
    "metadata": {},
    "execution_count": 43
  }
],
{
  "cell_type": "code",
```

```

"source": [
    "# replace the null data with null string\n",
    "data = df.where((pd.notnull(df)),)"
],
"metadata": {
    "id": "Cy6XeS8TfgMY"
},
"execution_count": null,
"outputs": []
},
{
    "cell_type": "code",
    "source": [
        "data['category'] = data['Category'].apply(lambda x: 1 if x == 'ham' else 0)"
    ],
    "metadata": {
        "id": "t2mBfH6Os9V-"
    },
    "execution_count": null,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "print(data)"
    ],

```

```
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "0l4ua6k8wDyY",
  "outputId": "52477809-0a33-44b7-e3b8-450fe6dd64bc"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "  Category                Message category\n",
      "0    ham  Go until jurong point, crazy.. Available only ...    1\n",
      "1    ham                Ok lar... Joking wif u oni...    1\n",
      "2    spam  Free entry in 2 a wkly comp to win FA Cup fina...    0\n",
      "3    ham  U dun say so early hor... U c already then say...    1\n",
      "4    ham  Nah I don't think he goes to usf, he lives aro...    1\n",
      "...    ...                ...    ...\n",
      "5567  spam  This is the 2nd time we have tried 2 contact u...    0\n",
      "5568  ham                Will ü b going to esplanade fr home?    1\n",
      "5569  ham  Pity, * was in mood for that. So...any other s...    1\n",
      "5570  ham  The guy did some bitching but I acted like i'd...    1\n",
      "5571  ham                Rofl. Its true to its name    1\n",
      "\n",
    ]
  }
]
```

```

    "[5572 rows x 3 columns]\n"
  ]
}
]
},
{
  "cell_type": "code",
  "source": [
    "data.drop(columns=['Category'])"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 423
    },
    "id": "QfCs45vrwOpZ",
    "outputId": "baab583b-e302-4a76-ba10-b31d2d6c9874"
  },
  "execution_count": null,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "
                Message category\n",
          "0    Go until jurong point, crazy.. Available only ...    1\n",

```



```

"1           Ok lar... Joking wif u oni...    1\n",
"2   Free entry in 2 a wkly comp to win FA Cup fina...    0\n",
"3   U dun say so early hor... U c already then say...    1\n",
"4   Nah I don't think he goes to usf, he lives aro...    1\n",
"...           ...    ...\n",
"5567 This is the 2nd time we have tried 2 contact u...    0\n",
"5568           Will ü b going to esplanade fr home?    1\n",
"5569 Pity, * was in mood for that. So...any other s...    1\n",
"5570 The guy did some bitching but I acted like i'd...    1\n",
"5571           Rofl. Its true to its name    1\n",
"\n",
"[5572 rows x 2 columns]"
],
"text/html": [
"\n",
"\n",
" <div id=\"df-159e9d41-81a3-43ec-91ae-f4fa388b2e4c\">\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>Message</th>\\n",
" <th>category</th>\\n",
" </tr>\\n",
" </thead>\\n",
" <tbody>\\n",
" <tr>\\n",
" <th>0</th>\\n",
" <td>Go until jurong point, crazy.. Available only ...</td>\\n",
" <td>1</td>\\n",
" </tr>\\n",
" <tr>\\n",
" <th>1</th>\\n",
" <td>Ok lar... Joking wif u oni...</td>\\n",
" <td>1</td>\\n",
" </tr>\\n",
" <tr>\\n",
```

" <th>2</th>\n",  
" <td>Free entry in 2 a wkly comp to win FA Cup fina...</td>\n",  
" <td>0</td>\n",  
" </tr>\n",  
" <tr>\n",  
" <th>3</th>\n",  
" <td>U dun say so early hor... U c already then say...</td>\n",  
" <td>1</td>\n",  
" </tr>\n",  
" <tr>\n",  
" <th>4</th>\n",  
" <td>Nah I don't think he goes to usf, he lives aro...</td>\n",  
" <td>1</td>\n",  
" </tr>\n",  
" <tr>\n",  
" <th>...</th>\n",  
" <td>...</td>\n",  
" <td>...</td>\n",  
" </tr>\n",  
" <tr>\n",  
" <th>5567</th>\n",  
" <td>This is the 2nd time we have tried 2 contact u...</td>\n",  
" <td>0</td>\n",  
" </tr>\n",  
" <tr>\n",  
" <th>5568</th>\n",

```

"   <td>Will ü b going to esplanade fr home?</td>\n",
"   <td>1</td>\n",
" </tr>\n",
" <tr>\n",
"   <th>5569</th>\n",
"   <td>Pity, * was in mood for that. So...any other s...</td>\n",
"   <td>1</td>\n",
" </tr>\n",
" <tr>\n",
"   <th>5570</th>\n",
"   <td>The guy did some bitching but I acted like i'd...</td>\n",
"   <td>1</td>\n",
" </tr>\n",
" <tr>\n",
"   <th>5571</th>\n",
"   <td>Rofl. Its true to its name</td>\n",
"   <td>1</td>\n",
" </tr>\n",
" </tbody>\n",
"</table>\n",
"<p>5572 rows × 2 columns</p>\n",
"</div>\n",
"   <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-
159e9d41-81a3-43ec-91ae-f4fa388b2e4c')\">\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.39.39.95.59 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",
"\n",
"\n",
" <div id=\"df-e093bac6-50d9-4592-a42e-f88448f428c1\">\n",
"   <button class=\"colab-df-quickchart\" onclick=\"quickchart('df-e093bac6-50d9-
4592-a42e-f88448f428c1')\">\n",
"     title=\"Suggest charts.\">\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",
"   <g>\n",
"     <path d=\"M19 3H5c-1.1 0-2 .9-2 2v14c0 1.1.9 2 2 2h14c1.1 0 2-.9 2-2V5c0-1.1-
.9-2-2-2zM9 17H7v-7h2v7zm4 0h-2V7h2v10zm4 0h-2v-4h2v4z\"/>\n",
"   </g>\n",

```

```
"</svg>\n",
"  </button>\n",
" </div>\n",
"\n",
"<style>\n",
" .colab-df-quickchart {\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-quickchart: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-quickchart {\n",
"   background-color: #3B4455;\n",
```

```

" fill: #D2E3FC;\n",
" }\n",
"\n",
" [theme=dark] .colab-df-quickchart: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",
" async function quickchart(key) {\n",
" const containerElement = document.querySelector('#' + key);\n",
" const charts = await google.colab.kernel.invokeFunction(\n",
" 'suggestCharts', [key], {});\n",
" }\n",
" </script>\n",
"\n",
" <script>\n",
"\n",
"function displayQuickchartButton(domScope) {\n",
" let quickchartButtonEl =\n",
" domScope.querySelector('#df-e093bac6-50d9-4592-a42e-f88448f428c1  
button.colab-df-quickchart');\n",
" quickchartButtonEl.style.display =\n",

```

```
"  google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"}\n",
"\n",
"  displayQuickchartButton(document);\n",
"</script>\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-159e9d41-81a3-43ec-91ae-f4fa388b2e4c
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-159e9d41-81a3-43ec-91ae-
f4fa388b2e4c');\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": 47
}

```

```
]
},
{
  "cell_type": "code",
  "source": [
    "X = data['Message']\n",
    "y = data['category']"
  ],
  "metadata": {
    "id": "iqj_bVt8BXXK"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)"
  ],
  "metadata": {
    "id": "9wIG6uBKBXYw"
  },
  "execution_count": null,
  "outputs": []
},
{
```

```

"cell_type": "markdown",
"source": [
    "Transform data into feature vactors"
],
"metadata": {
    "id": "bYoRgJYlhgM5"
}
},
{
    "cell_type": "code",
    "source": [
        "feature_extraction = TfidfVectorizer(min_df=1, stop_words = 'english',
lowercase=True)\n",
        "X_train_feature = feature_extraction.fit_transform(X_train)\n",
        "X_test_feature = feature_extraction.transform(X_test)\n",
        "\n",
        "y_train = y_train.astype('int')\n",
        "y_test = y_test.astype('int')\n"
    ],
    "metadata": {
        "id": "n4se9VrNgZdf"
    },
    "execution_count": null,
    "outputs": []
},
{

```

```
"cell_type": "code",
"source": [
    "print(X_train_feature) # each message gets score by Tfidfvectorizer"
],
"metadata": {
    "colab": {
        "base_uri": "https://localhost:8080/"
    },
    "id": "P-er683BgZbr",
    "outputId": "69d010e2-3ca0-416a-f591-40a2df86f0ac"
},
"execution_count": null,
"outputs": [
    {
        "output_type": "stream",
        "name": "stdout",
        "text": [
            " (0, 5818)\t0.22682143517864364\n",
            " (0, 2497)\t0.2442158912653505\n",
            " (0, 694)\t0.3171299579602537\n",
            " (0, 6264)\t0.1898892037332199\n",
            " (0, 5800)\t0.17558937755823417\n",
            " (0, 3262)\t0.33791755486732394\n",
            " (0, 2049)\t0.3034375179183143\n",
            " (0, 7300)\t0.24288153842988894\n",
            " (0, 2724)\t0.3544175987866074\n",
```

" (0, 354)\t0.3544175987866074\n",  
" (0, 7162)\t0.2550284465664535\n",  
" (0, 258)\t0.2379428657041507\n",  
" (0, 7222)\t0.2173884735352799\n",  
" (0, 5512)\t0.1898892037332199\n",  
" (1, 2555)\t0.3840709491751004\n",  
" (1, 3804)\t0.1902902346515268\n",  
" (1, 3932)\t0.24325511357721427\n",  
" (1, 4509)\t0.4028245991060671\n",  
" (1, 2440)\t0.33870544648398715\n",  
" (1, 3333)\t0.20665394084233096\n",  
" (1, 5650)\t0.360444144470318\n",  
" (1, 2335)\t0.2162321275166079\n",  
" (1, 6738)\t0.28986069568918\n",  
" (1, 6109)\t0.3239762634465801\n",  
" (1, 3267)\t0.2678713077029217\n",  
" :t:\n",  
" (4452, 2438)\t0.4574160733416501\n",  
" (4452, 7280)\t0.3968991650168732\n",  
" (4452, 3978)\t0.4574160733416501\n",  
" (4452, 3290)\t0.26370969643076225\n",  
" (4452, 3084)\t0.22948428918295163\n",  
" (4452, 2236)\t0.2676662072392096\n",  
" (4453, 3874)\t0.6064947019588056\n",  
" (4453, 4004)\t0.5244851817485773\n",  
" (4453, 6108)\t0.5975612693457145\n",

```

" (4454, 6113)\t0.4465347909835087\n",
" (4454, 6114)\t0.4465347909835087\n",
" (4454, 5149)\t0.43410473161397095\n",
" (4454, 5409)\t0.4079234999314281\n",
" (4454, 3249)\t0.3182708584577292\n",
" (4454, 2893)\t0.38087861810984514\n",
" (4455, 5815)\t0.5332274226200294\n",
" (4455, 3691)\t0.5541750775894743\n",
" (4455, 4660)\t0.4924788339394118\n",
" (4455, 6686)\t0.40745931976870786\n",
" (4456, 4518)\t0.5364209818026567\n",
" (4456, 6078)\t0.46545159250664164\n",
" (4456, 6467)\t0.48168628392630153\n",
" (4456, 5719)\t0.3276287995831882\n",
" (4456, 2236)\t0.31389751705425334\n",
" (4456, 3720)\t0.24023610815826446\n"
]
}
]
},
{
  "cell_type": "code",
  "source": [
    "model = LogisticRegression()\n",
    "model.fit(X_train_feature, y_train)"
  ],

```

```

"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/",
    "height": 74
  },
  "id": "6hqH3Z68gZZi",
  "outputId": "645a05ec-69c5-4213-bc2c-7978135ef368"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "LogisticRegression()"
      ],
      "text/html": [
        "<style>#sk-container-id-4 {color: black;background-color: white;}#sk-container-id-4 pre{padding: 0;}#sk-container-id-4 div.sk-toggleable {background-color: white;}#sk-container-id-4 label.sk-toggleable__label {cursor: pointer;display: block;width: 100%;margin-bottom: 0;padding: 0.3em;box-sizing: border-box;text-align: center;}#sk-container-id-4 label.sk-toggleable__label-arrow:before {content: '\\\" \";float: left;margin-right: 0.25em;color: #696969;} pre</div></div></div></div></div>"
      ]
    },
    "metadata": {},
    "execution_count": 52
  }
]

```



```
}
]
},
{
  "cell_type": "code",
  "source": [
    "pred_on_training_data = model.predict(X_train_feature)\n",
    "acc_on_training_data = accuracy_score(y_train, pred_on_training_data)"
  ],
  "metadata": {
    "id": "7y_rsWxGgY95"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "print(\"Accuracy on training data:\",acc_on_training_data)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "96PiPVjqnfKt",
    "outputId": "bff04298-2013-4b33-8581-e6d07e19dcd2"
```

```
},
"execution_count": null,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "Accuracy on training data: 0.9661207089970832\n"
    ]
  }
],
},
{
  "cell_type": "code",
  "source": [
    "pred_on_test_data = model.predict(X_test_feature)\n",
    "acc_on_test_data = accuracy_score(y_test, pred_on_test_data)"
  ],
  "metadata": {
    "id": "edEQgOKXnfHb"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "code",
```

```
"source": [
  "print(\"Accuracy on test data:\",acc_on_test_data)"
],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "HkmvJfY8TPT0",
  "outputId": "2271c2f9-721d-4874-c466-09af16077c04"
},
"execution_count": null,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "Accuracy on test data: 0.967713004484305\n"
    ]
  }
],
{
  "cell_type": "code",
  "source": [
    "input_your_mail = [\"Hi frnd, which is best way to avoid missunderstnding wit our\nbeloved one's?\"]\n",
```

```
"input_data_features = feature_extraction.transform(input_your_mail)\n",  
"prediction = model.predict(input_data_features)\n",  
"\n",  
"if(prediction[0] == 1):\n",  
"    print(\"Ham mail\")\n",  
"else:\n",  
"    print(\"Spam mail\")"  
],  
"metadata": {  
    "id": "TX_FT26foFZu",  
    "outputId": "4465a294-96db-439d-a2e5-2147527b3a84",  
    "colab": {  
        "base_uri": "https://localhost:8080/"  
    }  
},  
"execution_count": null,  
"outputs": [  
    {  
        "output_type": "stream",  
        "name": "stdout",  
        "text": [  
            "Ham mail\n"  
        ]  
    }  
]  
}
```

]

}