```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 1,
   "id": "0faff956",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:18.859702Z",
     "iopub.status.busy": "2021-12-21T16:51:18.858453Z",
     "iopub.status.idle": "2021-12-21T16:51:26.306435Z",
     "shell.execute_reply": "2021-12-21T16:51:26.305604Z",
     "shell.execute_reply.started": "2021-12-21T16:21:45.336780Z"
    },
    "id": "tPXrnm0Dn2ek",
    "papermill": {
     "duration": 7.479767,
     "end_time": "2021-12-21T16:51:26.306610",
     "exception": false,
     "start_time": "2021-12-21T16:51:18.826843",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "import pandas as pd\n",
    "import numpy as np\n",
    "import matplotlib.pyplot as plt\n",
    "%matplotlib inline\n",
    "import seaborn as sns\n",
    "import tensorflow as tf\n",
    "from tensorflow import keras\n",
    "from tensorflow.keras.models import Sequential\n",
    "from tensorflow.keras.layers import Dense,Flatten,Dropout\n",
    "from tensorflow.keras.optimizers import Adam\n",
    "from tensorflow.keras.models import load_model\n",
    "#from keras.utils import to_categorical\n",
    "#importing models\n",
    "from sklearn.model_selection import train_test_split\n",
    "from sklearn.preprocessing import LabelEncoder, MinMaxScaler\n",
    "import time\n",
    "import warnings\n",
    "from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score,
classification_report, confusion_matrix"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "id": "8674b4f2",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:26.369062Z",
     "iopub.status.busy": "2021-12-21T16:51:26.367918Z",
     "iopub.status.idle": "2021-12-21T16:51:26.370692Z",
     "shell.execute_reply": "2021-12-21T16:51:26.370116Z",
     "shell.execute_reply.started": "2021-12-21T16:22:00.467881Z"
    },
    "papermill": {
     "duration": 0.036674,
     "end_time": "2021-12-21T16:51:26.370869",
     "exception": false,
     "start_time": "2021-12-21T16:51:26.334195",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "from sklearn.preprocessing import StandardScaler"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "id": "11002db1",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:26.433348Z",
     "iopub.status.busy": "2021-12-21T16:51:26.432606Z",
     "iopub.status.idle": "2021-12-21T16:51:26.447538Z",
     "shell.execute_reply": "2021-12-21T16:51:26.448124Z",
     "shell.execute_reply.started": "2021-12-21T16:22:03.446706Z"
    },
    "papermill": {
     "duration": 0.049466,
     "end_time": "2021-12-21T16:51:26.448369",
     "exception": false,
     "start_time": "2021-12-21T16:51:26.398903",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "/kaggle/input/churn-modellingcsv/Churn_Modelling.csv\n"
     ]
    }
   ],
   "source": [
    "import os\n",
    "for dirname, _, filenames in os.walk('/kaggle/input'):\n",
    "    for filename in filenames:\n",
    "        print(os.path.join(dirname, filename))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "id": "ece44297",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:26.509075Z",
     "iopub.status.busy": "2021-12-21T16:51:26.507831Z",
     "iopub.status.idle": "2021-12-21T16:51:26.568633Z",
     "shell.execute_reply": "2021-12-21T16:51:26.567940Z",
     "shell.execute_reply.started": "2021-12-21T16:22:18.140727Z"
    },
    "id": "W6My8YVOpFYl",
    "papermill": {
     "duration": 0.091685,
     "end_time": "2021-12-21T16:51:26.568839",
     "exception": false,
     "start_time": "2021-12-21T16:51:26.477154",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "df = pd.read_csv('/kaggle/input/churn-modellingcsv/Churn_Modelling.csv')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 5,
   "id": "b994024f",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:26.645751Z",
     "iopub.status.busy": "2021-12-21T16:51:26.644842Z",
   "iopub.status.idle": "2021-12-21T16:51:26.671756Z",
   "shell.execute_reply": "2021-12-21T16:51:26.672337Z",
   "shell.execute_reply.started": "2021-12-21T16:22:19.827533Z"
  },
  "id": "J6epjyw1pFgc",
  "papermill": {
   "duration": 0.075692,
   "end_time": "2021-12-21T16:51:26.672546",
   "exception": false,
   "start_time": "2021-12-21T16:51:26.596854",
   "status": "completed"
  },
  "tags": []
 },
 "outputs": [
  {
   "data": {
    "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>RowNumber</th>\n",
"      <th>CustomerId</th>\n",
"      <th>Surname</th>\n",
"      <th>CreditScore</th>\n",
"      <th>Geography</th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Tenure</th>\n",
"      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>EstimatedSalary</th>\n",
"      <th>Exited</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>1</td>\n",
"      <td>15634602</td>\n",
"      <td>Hargrave</td>\n",
"      <td>619</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>42</td>\n",
"      <td>2</td>\n",
"      <td>0.00</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101348.88</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>2</td>\n",
"      <td>15647311</td>\n",
"      <td>Hill</td>\n",
"      <td>608</td>\n",
"      <td>Spain</td>\n",
"      <td>Female</td>\n",
"      <td>41</td>\n",
"      <td>1</td>\n",
"      <td>83807.86</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>112542.58</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>3</td>\n",
"      <td>15619304</td>\n",
"      <td>Onio</td>\n",
"      <td>502</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>42</td>\n",
"      <td>8</td>\n",
"      <td>159660.80</td>\n",
"      <td>3</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>113931.57</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>4</td>\n",
"      <td>15701354</td>\n",
"      <td>Boni</td>\n",
"      <td>699</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>39</td>\n",
"      <td>1</td>\n",
"      <td>0.00</td>\n",
"      <td>2</td>\n",
"    <td>0</td>\n",
"    <td>0</td>\n",
"    <td>93826.63</td>\n",
"    <td>0</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>4</th>\n",
"    <td>5</td>\n",
"    <td>15737888</td>\n",
"    <td>Mitchell</td>\n",
"    <td>850</td>\n",
"    <td>Spain</td>\n",
"    <td>Female</td>\n",
"    <td>43</td>\n",
"    <td>2</td>\n",
"    <td>125510.82</td>\n",
"    <td>1</td>\n",
"    <td>1</td>\n",
"    <td>1</td>\n",
"    <td>79084.10</td>\n",
"    <td>0</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>...</th>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
"    <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"      <td>...</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>9995</th>\n",
"      <td>9996</td>\n",
"      <td>15606229</td>\n",
"      <td>Obijiaku</td>\n",
"      <td>771</td>\n",
"      <td>France</td>\n",
"      <td>Male</td>\n",
"      <td>39</td>\n",
"      <td>5</td>\n",
"      <td>0.00</td>\n",
"      <td>2</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>96270.64</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>9996</th>\n",
"      <td>9997</td>\n",
"      <td>15569892</td>\n",
"      <td>Johnstone</td>\n",
"      <td>516</td>\n",
"      <td>France</td>\n",
"      <td>Male</td>\n",
"      <td>35</td>\n",
"      <td>10</td>\n",
"      <td>57369.61</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101699.77</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>9997</th>\n",
"      <td>9998</td>\n",
"      <td>15584532</td>\n",
"      <td>Liu</td>\n",
"      <td>709</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>36</td>\n",
"      <td>7</td>\n",
"      <td>0.00</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"     <td>42085.58</td>\n",
"     <td>1</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>9998</th>\n",
"     <td>9999</td>\n",
"     <td>15682355</td>\n",
"     <td>Sabbatini</td>\n",
"     <td>772</td>\n",
"     <td>Germany</td>\n",
"     <td>Male</td>\n",
"     <td>42</td>\n",
"     <td>3</td>\n",
"     <td>75075.31</td>\n",
"     <td>2</td>\n",
"     <td>1</td>\n",
"     <td>0</td>\n",
"     <td>92888.52</td>\n",
"     <td>1</td>\n",
"   </tr>\n",
"   <tr>\n",
"     <th>9999</th>\n",
"     <td>10000</td>\n",
"     <td>15628319</td>\n",
"     <td>Walker</td>\n",
"     <td>792</td>\n",
"     <td>France</td>\n",
"     <td>Female</td>\n",
"     <td>28</td>\n",
"     <td>4</td>\n",
"     <td>130142.79</td>\n",
"     <td>1</td>\n",
"     <td>1</td>\n",
"     <td>0</td>\n",
"     <td>38190.78</td>\n",
"     <td>0</td>\n",
"   </tr>\n",
"  </tbody>\n",
"</table>\n",
"<p>10000 rows × 14 columns</p>\n",
"</div>"
],
"text/plain": [
"      RowNumber  CustomerId    Surname  CreditScore Geography  Gender  Age  \\\n",
"0             1    15634602   Hargrave          619    France  Female   42   \n",
"1             2    15647311       Hill          608     Spain  Female   41   \n",
"2             3    15619304       Onio          502    France  Female   42   \n",
"3             4    15701354       Boni          699    France  Female   39   \n",
"4             5    15737888   Mitchell          850     Spain  Female   43   \n",
"...         ...         ...        ...          ...       ...     ...  ...   \n",
"9995       9996    15606229   Obijiaku          771    France    Male   39   \n",
"9996       9997    15569892  Johnstone          516    France    Male   35   \n",
"9997       9998    15584532        Liu          709    France  Female   36   \n",
"9998       9999    15682355  Sabbatini          772   Germany    Male   42   \n",
"9999      10000    15628319     Walker          792    France  Female   28   \n",
"\n",
"      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \\\n",
"0          2       0.00              1          1               1   \n",
    "1         1   83807.86              1         0              1   \n",
    "2         8  159660.80              3         1              0   \n",
    "3         1       0.00              2         0              0   \n",
    "4         2  125510.82              1         1              1   \n",
    "...     ...        ...            ...       ...            ...   \n",
    "9995      5       0.00              2         1              0   \n",
    "9996     10   57369.61              1         1              1   \n",
    "9997      7       0.00              1         0              1   \n",
    "9998      3   75075.31              2         1              0   \n",
    "9999      4  130142.79              1         1              0   \n",
    "\n",
    "      EstimatedSalary  Exited  \n",
    "0           101348.88       1  \n",
    "1           112542.58       0  \n",
    "2           113931.57       1  \n",
    "3            93826.63       0  \n",
    "4            79084.10       0  \n",
    "...               ...     ...  \n",
    "9995         96270.64       0  \n",
    "9996        101699.77       0  \n",
    "9997         42085.58       1  \n",
    "9998         92888.52       1  \n",
    "9999         38190.78       0  \n",
    "\n",
    "[10000 rows x 14 columns]"
   ]
  },
  "execution_count": 5,
  "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df"
 ]
},
{
 "cell_type": "code",
 "execution_count": 6,
 "id": "9253fe0a",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:51:26.735356Z",
   "iopub.status.busy": "2021-12-21T16:51:26.734599Z",
   "iopub.status.idle": "2021-12-21T16:51:26.746047Z",
   "shell.execute_reply": "2021-12-21T16:51:26.745490Z",
   "shell.execute_reply.started": "2021-12-21T16:22:20.155393Z"
  },
  "id": "spYXA5FApFpd",
  "papermill": {
   "duration": 0.044519,
   "end_time": "2021-12-21T16:51:26.746221",
   "exception": false,
   "start_time": "2021-12-21T16:51:26.701702",
   "status": "completed"
  },
  "tags": []
 },
"outputs": [
 {
  "data": {
   "text/plain": [
    "RowNumber          0\n",
    "CustomerId         0\n",
    "Surname            0\n",
    "CreditScore        0\n",
    "Geography          0\n",
    "Gender             0\n",
    "Age                0\n",
    "Tenure             0\n",
    "Balance            0\n",
    "NumOfProducts      0\n",
    "HasCrCard          0\n",
    "IsActiveMember     0\n",
    "EstimatedSalary    0\n",
    "Exited             0\n",
    "dtype: int64"
   ]
  },
  "execution_count": 6,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "df.isnull().sum()"
]
```

```
},
{
 "cell_type": "code",
 "execution_count": 7,
 "id": "a77ad880",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:51:26.815145Z",
   "iopub.status.busy": "2021-12-21T16:51:26.814458Z",
   "iopub.status.idle": "2021-12-21T16:51:26.834743Z",
   "shell.execute_reply": "2021-12-21T16:51:26.834175Z",
   "shell.execute_reply.started": "2021-12-21T16:22:20.475105Z"
  },
  "id": "0guxKKH0pFnv",
  "papermill": {
   "duration": 0.059489,
   "end_time": "2021-12-21T16:51:26.834906",
   "exception": false,
   "start_time": "2021-12-21T16:51:26.775417",
   "status": "completed"
  },
  "tags": []
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "<class 'pandas.core.frame.DataFrame'>\n",
      "RangeIndex: 10000 entries, 0 to 9999\n",
      "Data columns (total 14 columns):\n",
      " #   Column           Non-Null Count  Dtype  \n",
      "---  ------           --------------  -----  \n",
      " 0   RowNumber        10000 non-null  int64  \n",
      " 1   CustomerId       10000 non-null  int64  \n",
      " 2   Surname          10000 non-null  object \n",
      " 3   CreditScore      10000 non-null  int64  \n",
      " 4   Geography        10000 non-null  object \n",
      " 5   Gender           10000 non-null  object \n",
      " 6   Age              10000 non-null  int64  \n",
      " 7   Tenure           10000 non-null  int64  \n",
      " 8   Balance          10000 non-null  float64\n",
      " 9   NumOfProducts    10000 non-null  int64  \n",
      " 10  HasCrCard        10000 non-null  int64  \n",
      " 11  IsActiveMember   10000 non-null  int64  \n",
      " 12  EstimatedSalary  10000 non-null  float64\n",
      " 13  Exited           10000 non-null  int64  \n",
      "dtypes: float64(2), int64(9), object(3)\n",
      "memory usage: 1.1+ MB\n"
     ]
    }
   ],
   "source": [
    "df.info()"
   ]
  },
  {
   "cell_type": "code",
"execution_count": 8,
"id": "1cd8e18d",
"metadata": {
 "execution": {
  "iopub.execute_input": "2021-12-21T16:51:26.901220Z",
  "iopub.status.busy": "2021-12-21T16:51:26.900503Z",
  "iopub.status.idle": "2021-12-21T16:51:26.904837Z",
  "shell.execute_reply": "2021-12-21T16:51:26.904338Z",
  "shell.execute_reply.started": "2021-12-21T16:22:20.757730Z"
 },
 "papermill": {
  "duration": 0.039477,
  "end_time": "2021-12-21T16:51:26.904979",
  "exception": false,
  "start_time": "2021-12-21T16:51:26.865502",
  "status": "completed"
 },
 "tags": []
},
"outputs": [
 {
  "data": {
   "text/plain": [
    "Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',\n",
    "       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',\n",
    "       'IsActiveMember', 'EstimatedSalary', 'Exited'],\n",
    "      dtype='object')"
   ]
  },
   "execution_count": 8,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.columns"
 ]
},
{
 "cell_type": "code",
 "execution_count": 9,
 "id": "7139a754",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:51:26.969846Z",
   "iopub.status.busy": "2021-12-21T16:51:26.968861Z",
   "iopub.status.idle": "2021-12-21T16:51:26.972334Z",
   "shell.execute_reply": "2021-12-21T16:51:26.972800Z",
   "shell.execute_reply.started": "2021-12-21T16:22:21.014201Z"
  },
  "papermill": {
   "duration": 0.038679,
   "end_time": "2021-12-21T16:51:26.972995",
   "exception": false,
   "start_time": "2021-12-21T16:51:26.934316",
   "status": "completed"
  },
  "tags": []
  },
  "outputs": [],
  "source": [
   "list_drob=['RowNumber','CustomerId','Surname']"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 10,
  "id": "4855f44d",
  "metadata": {
   "execution": {
    "iopub.execute_input": "2021-12-21T16:51:27.040156Z",
    "iopub.status.busy": "2021-12-21T16:51:27.039360Z",
    "iopub.status.idle": "2021-12-21T16:51:27.042758Z",
    "shell.execute_reply": "2021-12-21T16:51:27.042194Z",
    "shell.execute_reply.started": "2021-12-21T16:22:21.249672Z"
   },
   "papermill": {
    "duration": 0.040751,
    "end_time": "2021-12-21T16:51:27.042899",
    "exception": false,
    "start_time": "2021-12-21T16:51:27.002148",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [],
  "source": [
   "df.drop(list_drob,axis=1,inplace=True)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 11,
  "id": "a12fd064",
  "metadata": {
   "execution": {
    "iopub.execute_input": "2021-12-21T16:51:27.117405Z",
    "iopub.status.busy": "2021-12-21T16:51:27.116246Z",
    "iopub.status.idle": "2021-12-21T16:51:27.120943Z",
    "shell.execute_reply": "2021-12-21T16:51:27.120389Z",
    "shell.execute_reply.started": "2021-12-21T16:22:21.487863Z"
   },
   "papermill": {
    "duration": 0.048486,
    "end_time": "2021-12-21T16:51:27.121087",
    "exception": false,
    "start_time": "2021-12-21T16:51:27.072601",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [
   {
    "data": {
     "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>CreditScore</th>\n",
"      <th>Geography</th>\n",
"      <th>Gender</th>\n",
"      <th>Age</th>\n",
"      <th>Tenure</th>\n",
"      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>EstimatedSalary</th>\n",
"      <th>Exited</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>619</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>42</td>\n",
"      <td>2</td>\n",
"      <td>0.00</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101348.88</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>608</td>\n",
"      <td>Spain</td>\n",
"      <td>Female</td>\n",
"      <td>41</td>\n",
"      <td>1</td>\n",
"      <td>83807.86</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>112542.58</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>502</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>42</td>\n",
"      <td>8</td>\n",
"      <td>159660.80</td>\n",
"      <td>3</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>113931.57</td>\n",
"      <td>1</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>3</th>\n",
"      <td>699</td>\n",
"      <td>France</td>\n",
"      <td>Female</td>\n",
"      <td>39</td>\n",
"      <td>1</td>\n",
"      <td>0.00</td>\n",
"      <td>2</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>93826.63</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
 "   <tr>\n",
 "     <th>4</th>\n",
 "     <td>850</td>\n",
 "     <td>Spain</td>\n",
 "     <td>Female</td>\n",
 "     <td>43</td>\n",
 "     <td>2</td>\n",
 "     <td>125510.82</td>\n",
 "     <td>1</td>\n",
 "     <td>1</td>\n",
 "     <td>1</td>\n",
 "     <td>79084.10</td>\n",
 "     <td>0</td>\n",
 "   </tr>\n",
 "  </tbody>\n",
 "</table>\n",
 "</div>"
],
"text/plain": [
 "   CreditScore Geography  Gender  Age  Tenure    Balance  NumOfProducts  \\\n",
 "0          619    France  Female   42       2       0.00              1   \n",
 "1          608     Spain  Female   41       1   83807.86              1   \n",
 "2          502    France  Female   42       8  159660.80              3   \n",
 "3          699    France  Female   39       1       0.00              2   \n",
 "4          850     Spain  Female   43       2  125510.82              1   \n",
 "\n",
 "   HasCrCard  IsActiveMember  EstimatedSalary  Exited  \n",
 "0          1               1        101348.88       1  \n",
 "1          0               1        112542.58       0  \n",
     "2        1           0      113931.57      1  \n",
     "3        0           0       93826.63      0  \n",
     "4        1           1       79084.10      0  "
    ]
   },
   "execution_count": 11,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.head()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 12,
 "id": "e058eec5",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:51:27.191614Z",
   "iopub.status.busy": "2021-12-21T16:51:27.187604Z",
   "iopub.status.idle": "2021-12-21T16:51:27.200921Z",
   "shell.execute_reply": "2021-12-21T16:51:27.200292Z",
   "shell.execute_reply.started": "2021-12-21T16:22:21.710619Z"
  },
  "papermill": {
   "duration": 0.04998,
   "end_time": "2021-12-21T16:51:27.201068",
     "exception": false,
     "start_time": "2021-12-21T16:51:27.151088",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "df=pd.get_dummies(df,columns=['Geography','Gender'])"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
   "id": "39fa2827",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:27.276656Z",
     "iopub.status.busy": "2021-12-21T16:51:27.275906Z",
     "iopub.status.idle": "2021-12-21T16:51:27.278812Z",
     "shell.execute_reply": "2021-12-21T16:51:27.279326Z",
     "shell.execute_reply.started": "2021-12-21T16:22:21.917300Z"
    },
    "papermill": {
     "duration": 0.048604,
     "end_time": "2021-12-21T16:51:27.279502",
     "exception": false,
     "start_time": "2021-12-21T16:51:27.230898",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [
    {
     "data": {
      "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>CreditScore</th>\n",
       "      <th>Age</th>\n",
       "      <th>Tenure</th>\n",
       "      <th>Balance</th>\n",
"      <th>NumOfProducts</th>\n",
"      <th>HasCrCard</th>\n",
"      <th>IsActiveMember</th>\n",
"      <th>EstimatedSalary</th>\n",
"      <th>Exited</th>\n",
"      <th>Geography_France</th>\n",
"      <th>Geography_Germany</th>\n",
"      <th>Geography_Spain</th>\n",
"      <th>Gender_Female</th>\n",
"      <th>Gender_Male</th>\n",
"    </tr>\n",
"  </thead>\n",
"  <tbody>\n",
"    <tr>\n",
"      <th>0</th>\n",
"      <td>619</td>\n",
"      <td>42</td>\n",
"      <td>2</td>\n",
"      <td>0.00</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>101348.88</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>1</th>\n",
"      <td>608</td>\n",
"      <td>41</td>\n",
"      <td>1</td>\n",
"      <td>83807.86</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>112542.58</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"    <tr>\n",
"      <th>2</th>\n",
"      <td>502</td>\n",
"      <td>42</td>\n",
"      <td>8</td>\n",
"      <td>159660.80</td>\n",
"      <td>3</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"      <td>113931.57</td>\n",
"      <td>1</td>\n",
"    <td>1</td>\n",
"    <td>0</td>\n",
"    <td>0</td>\n",
"    <td>1</td>\n",
"    <td>0</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>3</th>\n",
"    <td>699</td>\n",
"    <td>39</td>\n",
"    <td>1</td>\n",
"    <td>0.00</td>\n",
"    <td>2</td>\n",
"    <td>0</td>\n",
"    <td>0</td>\n",
"    <td>93826.63</td>\n",
"    <td>0</td>\n",
"    <td>1</td>\n",
"    <td>0</td>\n",
"    <td>0</td>\n",
"    <td>1</td>\n",
"    <td>0</td>\n",
"  </tr>\n",
"  <tr>\n",
"    <th>4</th>\n",
"    <td>850</td>\n",
"    <td>43</td>\n",
"    <td>2</td>\n",
"    <td>125510.82</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>79084.10</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>0</td>\n",
"      <td>1</td>\n",
"      <td>1</td>\n",
"      <td>0</td>\n",
"    </tr>\n",
"  </tbody>\n",
"</table>\n",
"</div>"
],
"text/plain": [
"   CreditScore  Age  Tenure    Balance  NumOfProducts  HasCrCard  \\\n",
"0          619   42       2       0.00              1          1   \n",
"1          608   41       1   83807.86              1          0   \n",
"2          502   42       8  159660.80              3          1   \n",
"3          699   39       1       0.00              2          0   \n",
"4          850   43       2  125510.82              1          1   \n",
"\n",
"   IsActiveMember  EstimatedSalary  Exited  Geography_France  \\\n",
"0               1        101348.88       1                 1   \n",
"1               1        112542.58       0                 0   \n",
"2               0        113931.57       1                 1   \n",
"3               0         93826.63       0                 1   \n",
"4               1         79084.10       0                 0   \n",
     "\n",
     "   Geography_Germany  Geography_Spain  Gender_Female  Gender_Male  \n",
     "0                  0                0              1            0  \n",
     "1                  0                1              1            0  \n",
     "2                  0                0              1            0  \n",
     "3                  0                0              1            0  \n",
     "4                  0                1              1            0  "
    ]
   },
   "execution_count": 13,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df.head()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 14,
 "id": "36d0392b",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:51:27.353934Z",
   "iopub.status.busy": "2021-12-21T16:51:27.348364Z",
   "iopub.status.idle": "2021-12-21T16:51:27.815899Z",
   "shell.execute_reply": "2021-12-21T16:51:27.815358Z",
   "shell.execute_reply.started": "2021-12-21T16:22:22.129135Z"
   },
   "papermill": {
    "duration": 0.506347,
    "end_time": "2021-12-21T16:51:27.816051",
    "exception": false,
    "start_time": "2021-12-21T16:51:27.309704",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
      "<AxesSubplot:xlabel='CreditScore', ylabel='Density'>"
     ]
    },
    "execution_count": 14,
    "metadata": {},
    "output_type": "execute_result"
   },
   {
    "data": {
     "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYgAAAEGCAYAAAB/+QKOAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90
bGliIHZlcnNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXBIWXMAAAsTAAALEw
EAmpwYAAAWi0lEQVR4nO3df7RdZX3n8ffHBCIKKoSoFLDBgrapWsWAOrV2VUcEOyW14hKsFVs6OI50deq
42qizKKWuruLMSGcqVmilw2CnBG11MtN0okjVjnUgQQUMGIiAEmA0/BgQFULgO3/sHTkcnntzudzcy95v95d9dcy95v
9Y63H2e/ex9vjv7cD93/zjPSVUhSdK4J026AEnS/GRASJKaDAhJUpMBIUlqMiAkSU2LJ13AXDnwwANr13AZSDnr+fLlky
5DkhaUK6644vaqWtaa94QJiOXLl7Nx48ZJlyFFC0qSb001z1NMkqQ1GRASJKaDAhJUpMBIUlqMiAkS
U0GhCSpyYCQJDUZEJKkJgNigs75V5dOugRJmpIBIUlqMiAkSU0GhCSpyYCQJDUZEJKkJgNCktRkQEiSmgwISV
KTASFJajIgJElNgwZEkmOTbE6yJcnqXvyvXJflKkh1JThibd3KS6/vHyUl6tMECIski4Bzgcl4sBzgOAFcFKSFWPdvg
```

28HfhvY8seAPw+8DLgaOD3k+w/VK2SpEcb8gjiaGBLVd1QVduBi4BVox2q6qaqugp4aGzZ1wGfrao7q+ou4L
PAsQPWKkkaM2RAHAzcPPJ8a982Z8smOTXJxiQbt23bNutCJUmPtqAvUlfVeVW1sqpWLlu2bNLlSNITypABc
Qtw6MjzQ/q2oZeVJM2BIQNiA3BEksOS7A2cCKyd4bLrgWOS7N9fnD6mb5Mk7SaDBURV7QBOo/vFfi1wcV
VtSnJmkuMBkhyVZCvwJuDcJJv6Ze8E/pAuZDYAZ/ZtkqTdZPGQK6+qdcC6sbbTR6Y30J0+ai17PnD+kPVJkqa
2oC9SS5KGY0BIkpoMCElSkwEhSWoyICRJTQaEJKnJgJAkNRkQkqQmA0KS1GRASJKaDAhJUpMBIUlqMiAkS
U0GhCSpyYCQJDUZEJKkJgNCktRkQEiSmgwISVKTASFJajIgJElNBoQkqcmAkCQ1GRCSpCYDQpLUZEBIkpoMC
ElSkwEhSWoyICRJTQaEJKlp0IBIcmySzUm2JFndmL8kyZp+/mVJlvfteyW5IMnVSa5N8t4h65QkPdpgAZFkEX
AOcBywAjgpyYqxbqcAd1XVAYCp4ODPTQiIw4+5uAJVX1QuClwDt2hockafcY8gjiaGBLVd1QVduBi4BVY31WARf005
8EXpMkQAFPTbIY2AfYDtwzYK2SpDFDBsTBwM0jz7f2bc0+VbUDuBtYShcW3wduA74N/IequnP8BZKcmmRj
ko3btm2b+y2QpD3YfL1IfTTwIPBjwGHAv03y3PFOVXVeVa2sqpXLli3b3TVK0hPakAFxC3DoyDoPyMe+rZmn/50
0tOBO4C3AP+rqh6oqu8CXwJWDlirJGnMkAGxATgiyWFJ9gZWA+vHc9ND9VkLnNxPwcBcWlVFd1rp1QBJngq8CBJngq8HP
jGgLVKksYMFhD9NYXTgPXAtcDFVbUpyZlJju7fQxYmmQL8G5g562w5wD7tlEFzR/WVVXDVWrJOnRFg+5
8qpaB6wbazt9ZPo+ultax5e7t9UuSdp95utFaknShBkQkqQmA0KS1GRASJKaDAhJUpMBIUlqMiAkSU0GhCS
pyYCQJDUZEJKkJgNCktRkQEiSmgwISVKTASFJajIgJElNBoQkqcmAkCQ1GRCSpCYDQpLUZEBIkpoMCElSkwE
hSWoyICRJTTMKiCR/m+QXkxgokrSHmOkv/I8AbwGuT/LHSZ4/YE2SpHlgRgFRVZdU1a8CLwRuBi5J8uVJb
kryELlCRNxoxPGSVZCrwd+E3gq8B/oguMzw5SmSRpohbPpFOSTwHPBy4EfqmqbutrUmycajiJEmTM6OAA
P68qtaNNiRZUlX3V9XKAeqSJE3YTE8xfqDR9pW5LESSNL/s9A0iyTPpvrdo3yQvANLPehqw38C1SZImaLpTT
K8F3gYcAnxwpP1e4H0D1SRJmgd2GhBVdQFwQZITqupTu6kmSdI8MN0ppl+rqr8Clif5d+Pzq+qDjcUkSU8A05
1iemr/uP/QhUiS5pfpTjGd2z/+0e4pR5I0X0x3iunsnc2vqt+Z23IkSfPFdKeYrt4tVUiS5p2ZfIpJkrQHm
u4U039a1e9O8hmgxudX1dGDVSZJmqjpTjFd2D/+h6ELkSTNL9OdYrqqf/xCkr3p/hFQAVuqavt2qE+SNCEzut13
kl8EPgrcSPd9TIcl+ddV9dkhi5MkTc5Mvw/iz4BXVdUNAEl+HPg7wICQpCeomX7d92/o9snIbpvkkMq+
A2S/gmThgsh30W0I3sWGDAYI+W6ilbQTO6eOgPGOa6r5SX4/+XX9/FbPdoGx8lQTZAH6XwT6NrsZeUcn31hFYZgNuWH7+k5mn8aqA+xfmd6UzXF7
Hd7nfd2tZn8Fz4C5bE6vf9bIRmRRZQb3w9Y1gPBqw1N9H+rpDMq35e0mRaXNAsk/Z3JAoDi+cyQO5MQKSlv
Qn+GQXmLHQrwWdV7GQ8ZpvV+JDvZOcfZvHxn+K+tRN+L3+TNjrWMjFBfSIWkPswwAbn6P+v5Nt5bB6+yy
JSHtmnxy9k+n88dgX3Rk5AtqDk97Bg0qRJUeuFHNhRMYYRE3Z+BiRYWsYXVkTsl/9odQEq8wIgCrbw
kLG9odrrfMeyTzTnkEa5hcS6rYBiFNqXWmQLGj9k/+3WDojIm5zxaeUPh/K3/GbI8vx7+WzOfx3sxzp+
5Y7ROn0Z6mgfrxPxp4+9Z8q5AhFG+B7OvGnxQUpTfNMA4mGhPk6Yas/4ADzgd4tsqSVcbyUdvOxg9dqbaN
JrgcK9H0gsEtYQqDY7w/4A7gVChBK4i9mvtpMZvUpNJUxeWzC8onnDpW/aQzYQ+5qrkqRtmKKzl7+zTGBF
twpNPjMjAAk1U+ZOTlUkBuGzfSYk7sDmr/ZD3bHpZMF8U0TgjJq8gSd1Fk6JXEMw3SDqA3DjNkGPlVH5

vK2qvjlgnZKkhiEDYgNwRJLDkuwNnAisHeuzlu4iNMAJwKVVVUmeAfwdsLqqvjRgjZKkKQwWEP01hdOA9cC
1wMVVtSnJmUmO77t9DFiaZAvwbmDnrbCnAYcDpyf5Wv945lC1SpIebfGQK6+qdcC6sbbTR6bvA97UWO4
DwAeGrE2SNL15fZFakjQ5BoQkqcmAkCQ1GRCSpCYDQpLUZEBIkpoMCElSkwEhSWoyICRJTQaEJKnJgJAkN
RkQkqQmA0KS1GRASJKaDAhJUpMBIUlqMiAmJpMuQJKmZUBIkpoMCElSkwEhSWoyICYlXoOQNL8ZEJKkJg
NCktRkQEiSmgwISVKTATFhm269e9IlSFKTATFhF11+86RLkKQmA2JivM1V0vxmQEiSmgyISekPIJ7kgYSkeWr
QgEhybJLNSbYkWd2YvyTJmn7+ZUmWj8x7b9++OcnrhqxzMrpk2P5gTbgOSWobLCCSLALOAY4DVgAnJVkx
1u0U4K6qOhw4GzirX3YFcLLw08CxwEf69T3zSbrJBrB+47qOBLVV1A0CSi4BVwDUjfVYBZ/TTnw
Q+nCR9+0VVdT9wY5It/fq+PEShP9zxQz5z02fY/uD97LvxOva643sUM/jLvh7uUyP/fXh2PartR7ZeBfwKL/7Ge
aw/e02jQx4xXFNGL2r3q3wswzllji6Kz+jfZYLqMZc3v7en89j23fwZ5muKQqYr8DHXPruN/d5Rz2PH0qfNatn5
6AVLX8BPLf2pOV/vkAFxMDB6D+dW4GVT9amqHUnuBpb27f9nbNmDx18gyanAqf3Te5Nsnk2hT3rKk5629
zP3PmLRQ7D8O7NZw+zcutdJ/NgD9+y+FxzAnQ/u4IBFQ76Nhuc2zA+7cxvu3A/u2nfu1/vg9x5k0X67/2THQ
/c/9L3tt22/bpaL//hUMxb0O6qqzgPOm3Qds5Vk451VKyddx+ORZOOtDzzgNkyY2zA/JNm4/fbtC3obRg15kf
oW4NCR54f0bc0+SRYDTwfumOGykqQBDRkQG4AjkhyWZG+6i85rx/qsBU7qfrBFa4MT+LqfDgCO
AywesVZI0ZrBTTP01hdOA9cAi4Pyq2pTkTGBjVa0FPgZc2F+EvpMuROj7XUx3QXsH8K6qenCoWidowZ4eG+
E2zA9uw/zwRNiGH0k99ts+JEl7AD9JLUlqMiAkSU0GxATsagiSSUhyU5Krk3wtyca+7YAkn01yff9z/749Sf5zX/
9VSY4cWc/Jff/rk5w80v7Sfv1b+mUf98e5kpyf5LtJvj7SNnjNU73GHG7DGUlu6ffF15K8fmRecwiaqd5T/U0il/
Xta/obRqYd5mYW23Bokn9Ick2STUl+u29fMPtimm1YUPtizlWVj934oLtg/03gucDewJXAinlQ103AgWNtHw
RW99OrgbP66dcDf0/3MdaXA5f17QcAN/Q/9++n9+/nXd73Tb/scXNQ86uAI4Gv786ap3qNOdyGM4D3NPq
u6N8vS4DD+vfRouneU8DFwIn99EeBd/bT/xr4aD99IrDmcWzDIcCR/fR+wHV9rQtmX0m4KB++iBgcz99LnDSeD/gJODckfZZz+7aDgG+MtD+i3+OsezmP/OU6eM1T
vcYcbsNUv5Qe8V6hu0PwFVO9p/pfrpcDi8ffezXj7orENC3pfPN6Hp5h2v9YQJI8aRm
QCCvhMkivSDWEC8Kyquq2f/r/As/rpqbZhuvatjfYh7I6ap3qNuXRaf/rl/JHTJo91G5YC/6+qdjS24RHD3AA7h
7l5XPrTIy8BLmOB7ouxbYAFui/mggGhnV5ZVUfSjb77riSvGp1Z3Z83C+qe6N1R80Cv8WfATwAvBm4D/uMc
r38QSfYF/gb4N1X1iEHGFsq+aGzDgtwXc8WA2P3m5TAiVXVL//O7wKfoRs/9Tn+qhP7nd/vuU23DdO1LJ22dbb9TGp1Z3Z83C+qe6N1R80Cv8WfATwAvBm4D/uMc
fwu6oearXmBNV9Z2qerCqHgL+nG5fzGYb7gCekW4Ym/FtmGqgVm1lJshfdL9a/qqq/7ZsX1L5obcNC3BdzyYD
Y/WYyBMluleSpSfbbOQ0cA3ydRw6FcjLdeVn69rf1d6O8HLi7P8xfDxyTZP/+UPwYuvOvtwH3JHl5f7f7W
Ndd2R81Tvcac2PkLr/cGun2x83VbQ9A031P9X9T/QDeMzXitUw1zM5t6QzcqwrVV9aGRWQtmX0y1DQttX
8y5SV8E2RMfdHdxXEd3t8P750E9z6W72+JKYNPOmujOg34OuB64BDigbw/dl0F9E7gaWDmyrt8AtvSPXx9
pX0n3P9c3gQ8zBxfhgL+mO+x/gO6c7im7o+apXmMOt+HCvsar6H55HDTS//19PZsZuRNsqvdUv28v77ftE8C
Svv3J/fMt/fznPo5teCXdqZ2rgK/1j9cvpH0xzTYsqH0x1w+H2pAkNXmKSZLUZEBIkpoMCElSkwEhSWoyICRJT
QaE9khJnp3koiTf7IcXWfkebNc139JckI//RdJVvTT7xvr9/5+pNCr+pFBX/b4t0QazmBfOSrNV/2Hoj4FXFBVJ/
ZtP0M3js91/fPF9fC4OTNWVb858vR9wB/163sF8C/oRgy9P8mBdKN9Pp7tmFWN0kx5BKE90S8AD1TVR3c2
VNWVwKIk/5hkLXBNkkVJ/n2SDf1f1f/e+AH32fwYfTjfl/CfDMnetJ8vkkK5P8MbBPf6TwV3Qjjd5eVff3r3d7Vd3
aL3NUkn9KcmWSy5Psl+TJSf4y3XcgfDXJL/R9355kbZJLgc/1n4I/v1/uq0lW7Z5/Qu0JPILQnugFwBVTzDsSeEF
V3ZhuVNu7q+qoJEuALyX5DN1In8+n+06AZwHXAOePrqSqVic5rapeDD8aBO70JNfRfeJ3TVV9oR+OYQ3w5
qrakORPpwA+B3+5WUy9M8pN0I+3uPAV2JPCiqrozyR/RDc3wG0meAVze2E8f
A7xo5/UFukHUjqD7kp+/rqoHgVv7v+anVV7v+anVVX3Jnkp8HN0RzBr0n3b2BXAbVW1oe93D0CSVwJ/2rd9I8m3gJ0
B8dmqunOkxuxOTvKd//mTgDcC1s9t86WEGhPZEm3h40huLTlNQzbFsbhbN3e9Nkmb4eBs
ojmemM1/jGqto8m5qk6XgNQnuiS4ElefiLkUjyIrq/7ketB96Zbhhokjwv3Wi3XwTe3F+jOIjuiKDlgZFln5/kiJF5
Lwa+Rf+NaEmO6vvtl27I538EfnXn69IdFbRCYD3wW/2Fd5K8Zlb/BtIueQShPU5VVZI3AH+S5HeAH9J9i+v/GGv6QDdZc3/ePMlPp5KTcXF4+S5PeA++i+cvXTY1
3/gu7rQL/S/wLeBvwy3R1Qr6a79vBt4MtTvNR5wFVJvgJ8CPjT/jrBDrqRO0+tqu1J3tzP24fu+sM/Bz4C/Fl/pL
EDeHt/99P4a/wh8Cf96zwJuJHubinpcXM0V0lSk6eYJElNBoQkqcmAkCQ1GRCSpCYDQpLUZEBIkpoMCElS0/
8HJ0txqYVxJvwAAAAASUVORK5CYII=\n",

"text/plain": [

```
       "<Figure size 432x288 with 1 Axes>"
      ]
     },
     "metadata": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
    "sns.kdeplot(df['CreditScore'], shade=True)\n",
    "sns.kdeplot(df['Age'], shade=True)\n",
    "sns.kdeplot(df['Balance'], shade=True)\n",
    "sns.kdeplot(df['EstimatedSalary'], shade=True)\n",
    "sns.kdeplot(df['Tenure'], shade=True)\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 15,
   "id": "e618512d",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:27.881579Z",
     "iopub.status.busy": "2021-12-21T16:51:27.880941Z",
     "iopub.status.idle": "2021-12-21T16:51:27.890915Z",
     "shell.execute_reply": "2021-12-21T16:51:27.891473Z",
     "shell.execute_reply.started": "2021-12-21T16:22:22.601501Z"
    },
   "papermill": {
    "duration": 0.044285,
    "end_time": "2021-12-21T16:51:27.891654",
    "exception": false,
    "start_time": "2021-12-21T16:51:27.847369",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [],
  "source": [
   "stand= StandardScaler()\n",
   "for column in ['CreditScore','Age','Balance','EstimatedSalary','Tenure']:\n",
   "    df[column] = stand.fit_transform(df[column].values.reshape(-1,1))"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 16,
  "id": "107b37b8",
  "metadata": {
   "execution": {
    "iopub.execute_input": "2021-12-21T16:51:27.958659Z",
    "iopub.status.busy": "2021-12-21T16:51:27.958024Z",
    "iopub.status.idle": "2021-12-21T16:51:28.378377Z",
    "shell.execute_reply": "2021-12-21T16:51:28.377793Z",
    "shell.execute_reply.started": "2021-12-21T16:22:22.615074Z"
   },
   "papermill": {
    "duration": 0.454954,
    "end_time": "2021-12-21T16:51:28.378530",
    "exception": false,
    "start_time": "2021-12-21T16:51:27.923576",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
      "<AxesSubplot:xlabel='CreditScore', ylabel='Density'>"
     ]
    },
    "execution_count": 16,
    "metadata": {},
    "output_type": "execute_result"
   },
   {
    "data": {
     "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYIAAAEGCAYAAABo25JHAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90b
GliIHZlcnNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXBIWXMAAAsTAAALEwE
AmpwYAABHq0lEQVR4nO3deXydZ3ng/d/1nH3RvlvyIlm2Y8d2NmcnJBACoVCWwjsspS1lCu106Ke8Zaaly0s
ppZ22hLbTQjtQhoHSThMgQFNIcBKyOSROvO+bbdMva9+3s6/3+cSRbto6ko+XoyNL1/XxkS+e5n+e5dCSd69
y7GGNQSim1eImFDkAppVRhaSJQlVThOBUkqtcpoIlFJqldNEoJRSq5y90AHMVWVlpdmwYUOhw1BKevv
KgQMHBowwxVdmOXXeJYMOGDezfv7/QYSil1HVFRC5NdQaucJgKlFlFrlNBEopdQqp4lAK
aVWOU0ESim1ymkiUEqpVU4TgVJKrXKaCFROekI9hQ5BKZUnmgjUrDoCHbz6e2lK9hV6FCUUnmgjUDN6o
X2FwB46uJTBY1DKZUfmgjUrJ5te5YH7tbdhQ5FKZUHmgjUrM6PnOe+7bz1e2+lK9hV6FCUUnmgiUDN6o
X2FwB46uJTBY1DKZUfmgjUrJ5te5YH1z3I7tbdhQ5FKZUHmgjUrM6PnOe+hvvoCHYUOhSlVB5oIlAziqfiBON
B6v31xJIxwolwoUNSSi0yTQRqRn3hPsrd5VhiUeGpoCeso4eUWmk0EagZ9YZ7KXOXAVDmLqMv3FfgiJRS
+FjJDZS2ICUUotOE4Ga0VhsDJ/DB2QSwhstMARK
```

aUWmyYCNaOx+JVE4Hf4GYuPFTgipdRi00SgZhSIB66qEQTiAYwxBY5KKbWYNBGoacVSMZLpJC6bCwC7Zcd
pcxJO6jITSq0kmgjUtMZiY/idfkTk8mN+h1/7CZRaYTQRqGmNxkYvNwtN8Dv8OnJIqRVGE4Ga1lh8DL/Df9Vj
OnJIqZVHE4GaVrYagdfhZTSuiUCplUQTgZrWaHwUr8N71WN+h5+xmA4hVWol0USgpjUaG8VrvzoReOweb
RpSaoXJayIQkYdF5IyItIjIZ7IcXyciz4vIIRE5KiI/l8941NxkSwRuu5tAPFCgiJRS+ZC3RCAiNuArwNuBbcCHRGTb
NcX+CPiOMeYW4IPAP+QrHjV3gXgAj8Nz1WMeu4dgIligiJRS+ZDPGsEdQIsx5oIxJg48Crz7mjIGKB7/vATQNY
6XkVAihNvmvuoxj92jNQKlVph8JoJ6oH3S1x3jj032OeAjItIBPAn8VrYLicgnRGS/iOzv7+/PR6wqi0AigMd+dY3
AbXdrjUCpFabQncUfAr5pjGkAfg74tohMickY8zVjzC5jzK6qqqolD3K1CsVDuO1TawShRKhAESml8iGfiaATW
Dvp64bxxyb7z8B3AIwxrwJuoDKPMak5CCVDU2oEHruHYFxrBEqtJPlMBPuATSLSKCJOMp3BT1xTph14K0xpg14EEBE
tpJJBNr2s0yEElNrBG6bm1BSawRKrSR5SwTGmCTwSWA3cIrM6KATIvJ5EXnXeLFPAx8XkSPAvwEfNbrG8bIR
SoTw2KbWCMIJXX1UqZXEns+LG2OeJNMJPPmxz076/CRwbz5jUPMXToS1j0CpVaDQncVqmUqbNNFkdEoi
sFt2jDHEU/ECRaaUWmyaCFRWkWQEl92Fdc0gLhHB6/BqrUCpFUQTgcoqGA9OGTE0QWcXK7WyaCJQWW
UbOjpB+wmUWlk0EaisQvGZE4HOJVBq5dBEoLIKJqZvGtJlJpRaWTQRqKyyLTg3wWVz6VwCpVYQTQQqq3A
yjMvuynrMZXMRTmoiUGql0ESgsgonwrhsMyQCrREotWJoIlBZhZNhYz6zAqVWEE0EKqvZagShZGiJo1FK5YsmApVVOBHGbXdnPaZNQ0qtHJoIVFbhZBiX3ZX1mMuuNQKlVhJNBCqrcDKM2+bOekybhpRaOTQRqKwiyQhu2/SJQJuGlFo5NBGorCLJCC7H9E1D4UQYnfVdqZVBE4HKKjJD05BNbFjEQjQZXeKolFL5oIlAZTVT0xCAx+HROw8ptUJoIlBZzdQ0BOC2uYkkIksYkVIqXzQRqKzCyfCMNYIRCPWCJRaKTQRqCkiyQhOS2sESq0WmgjUFLn2OUwCFw2F+GE9hEotRJoIlBTzzJ5TzLQpzQTLFZq5dBEoKbIJRHo8FGlVg5NBGqKcDKsTUNKrSJ5TQQi8rCInBGRFhH5zDRl/pOInBSRElyLyf/M
Zj8pNrjOLtWlIqZUh+4Lzi0BEbMBXgIeADmCfiDxhjDk5qcwm4PeBe40xwyJSna94VO50rSGlVpd81gjuAFqM
MReMMXHgUeDd15T5T5OPAVY8wwgDGmL4/xqBzlvMSErjWk1IqQz0RQD7RP+rpj/RnrpJXRB7O
diER+YSI7BeR/f39/XkVO03Iafio3UUkpYlAqZWg0J3FdmAT8ADwIeCfRKT02kLGmK8ZY3YZY3ZVVVUtbYSrU
DQ1/X7FE1w2l25VqdQKkc9E0AmsmsbUATxhjEsaYi8BZMolBFZJqmnqhdWAWDSP4F0AwnVCGkXRnIrjJDRoFeaiwFmEBzspCaKJbnFA
0YM5ivmFRuckkETstJIp0glU4tUVRKqXzJ2/BRAGPMk8CT1zz22UmfG+B3x3/j/UMpBIJ0ibNHZr5l8NEcnUClJRf
JZviaJTSuVDoTuL1TITTUZx2V2IyKxldS6BUiuDJgJ1lUgygtvmzqmsy66JQKmVQBOBukokGcClGWCJRaG
TQRqKvkMnR0giYCpVYGTQTqKpoIlFp9NBGoq+Sy4NwwEp82p6w0ptQLklAhE5Psi8g4R0MVjnvTiz34zwuWyztAErREot
TLk+sL+D8CHgXMi8i8hcisiWPMakCiqRm37h+gtNyEk3p7GmF8EbgVaAa+WdF6BNnJYYmAqVWgpyWmBCRHgD8D
QLW05tJH4LTrekNKrQQ5N/WISAXwyPmNATI+hLl2X+R4P/GL8GbrtfzCFAgb/Y9AE4FSK0
P2+M6R4/9JiI7M9XcGrpRZIRHLbcKnl/Om1P3LVZqBch1raFfAeuIRPSbsa7rK1bQ4XvQpSvFMoa
uTUNfyPLYq4sZiFoewonZN66foKOGlFoZZqwRiEgtme0lPSJyq7SMzM1zbKoAI+sKpa7SnMq6bC76kyt769
Czw2f58qEv0zLSwtbyrXzq1k+xtnjt7CcqdR2ZrUbwNuARMruL/TXwpfGP+IL+hqULQmcVXvNr1Kh/7yce
o9dXy8R0fp9hZzIef/DAnB08WOjSlFtWMNQJjzLeAb4nI+4wxjy9Nn73pd/lN276DbaUZ6bN
NBQ1UO2t5ree+y2+/67vU+IqKXCUSi2OGWsEIvKR8U83iMvXPuxBPGpJRZOhnMfPrpC9y02xvBHP/sjHt7w
8OUkMGFX7S52VO7gS/u/VKDolFp8szUNTWw95QeKsnyoFSaSjOCyVneN4JlLzzASHEt69+S9fh7m9/L8+3
Pc2743BJHplR+zNY09NXx//9kacJRhTaXtYZWYo0gbdL83aG/432b3oc1zdJaXoeXh9Y/xFePfpVH7n9kiSNUa
vHluujcX4lIsYg4ROjROSnItIlt/qdlIrSC5bFw/wW1zE02urLWGXmh/AbvYubHixhnL3d9wP690vkJXsGtpAlMqj3Kd
R/BWY8wY8E4yaw01A/89X0GGpwll2ncXpNHQfgeOPw/HvQ/dRMCZvt/vnk//Mg+sfnHXPZq/Dy91r7uZ7Z7+
Xt1iUWiq5ziyeKPcO4LvGmNFcNjdX15+51AictszqoSOz9R9j/DXB4oHR9ZhbL4AWwbPDQn8C
298Ai3rdtrI2W4RY+sfMTOZW/r+E+/vbA3/KbN/8mdivXPyWllp9cf3t/JCKngQjwX0SkClhZbQIKYwyxVCznR
GC37FhYJNKJnPsVcnLyCfjxp6HuJnjL56B03eQgofcXPXf5lMe1kWbz/XPf5641d+1Gwcrtevb+eMnc
Ze7v38ob6NyxKDEoVQq7LUH8GuAfYZYxJAAHg3fkMTC2tWCqGw+LAIrnNM8zbyKELP4Hv/Qrs+lX4f/xe/s3ruPXW/+vbeTF8fhQ8Dk/R4AO1AOfdJv+UZrW2AAnEb
Z3hXSbC8yKcycxAU+IRbKAeT5QbYOWHyJVqTWAAxyJeNNNlsxBeWT9XE+MYW1rYDUpxo8xXxNSo1kCv+2L8i/sXqjn

/cBb8xOSKpRIMoLb7p7TOW6be3H6CVIJ+N5HoX0v/NwjUL115vIONzzwGQj1wdN/tODbHx04iiUW64rWz
V54kjtq7+DFjheJpWILjkGpQsl11NC3ySw18Qbg9vEPXXV0hZnL8hIT3PZFSATpNHzvYxDsgzd/Flw5TlGx7PD
G34UTP8w0Ey3AM63PcGv1rXPu6yhxlbCuaB17u/Yu6P5KFVKufQS7gG3G5HG4hiq4uaw8OmFRRg498//Bc
Cu85U/m3t7vKoJ7Pgn/8duw7i7wlM359sYYnm17ll/b8WtzPhfg5uqb2d26m/vX3j+v85UqtFybho4DtfkMRB
XefGoELpuLcHIBfQTHvgcnfwgP/P78O31rd0LDHfDTz8/r9AujF4ilYnNuFppwS/Ut7OncQ2oR+iqUKoRcE0Elc
FJEdovIExMf+QxMLb25zCGYsKDZxYPn4cn/lmneybU5aDo3fQhO/AD6z8z51Jc6XmJH5Y55D4Gt9FRS7Czm+
ODxeZ2vVKHl2jT0uXwGoZaHJa0RpNPwg1+HHf8Jypvmfv613MWw/X2ZjuNf/O6cTn2x40XuWXPPgm6/vXI
7L7a/yE1VNy3oOkoVQq7DR18kM6PYMf75PuBgHuNSBTDfRBBJzKNG8PrXIBGBG94x93Ons+XnoOtQZiZyjs
KJMCcHT7K1fJZRSrPYUbmDlzpeWtA1lCqUXEcNfRz4HvDV8YfqgR/mKSZVIJFkJOf9iifMq2ko0Asv/A+4+7/C
HOYszMrmzMw2fuEvcj7lQO8BGosb5zxs9lobSzfSEexgMDK4oOsoVQi5/hX+V+BeYAzAGHMOqM5XUKowI
skITvscE4E1j0Tw7Oeg+UEoycOWj5vfBm17c+4reLXr1Sl7DsyH3bKztXwrr3brVt7q+pNrIogZY+ITX4iIHdChpC
vMfGsEc5pZ3HMMzu2GHR+YY3Q5srth88Ow9x9yKv5K1ytsrVhYs9CELeVbeLVLE4G6/uSaCF4UkT8gs4n9Q
8B3AZ1Xv8KEEqE5rxk0587iZ/8EdrwfnN45RjcHW96eWa00PDRjsZHoCF2hLjYUb1iU226r2Mbe7r3odBt1vc
k1EXwG6AeOAb8OPAksfF6/WlbmNaFsLmsNtb+eqRFsenge0c2BpwzW3gUH/3nGYgd6D7C5bPOirRxa660l
ZVK0jrUuyvWUWiq5jhpKk+kc/k1jzPuNMf+ks4xXnnByfjOLc64RPP/nmdrAIq0WOqPNb4P9/zszTHUar/W8
RnNp86LdUkTYWr6VfT37Fu2aSi2F2TavFxH5nIgMAGeAM+O7k302l4uLyMMickZEWkTkMzOUe5+IGBHR9
YsKKJKY5/DRXDqLuw5D7wnY+OD8gpuryi2ZUUQXX5y2yP6e/WwpW3hH8WSbyzZrP4G67sxWI/h/yYwWu
t0YU26MKQfuBO4Vkf93phNFxAZ8BXg7sA34kIhsy1KuCPht4LV5xK8W0XxqBG6bO7emoZcegW3vWpraAG
SWq970Vtj3T1kPB+IB2gPtbCjZsKi3vaH8Bg70HtB+AnVdmS0R/BLwIWPMxYkHjDEXgI8AvzzLuXcALcaYC+M
jjh4l+x4Gfwr8JbrRTcGFk+E5LzHhsufQNDTcCq17YNPb5h/cfDQ+ABdegtDAlENH+o/QVNK06DuLVXoqcdqc
XBi9sKjXVSqfZksEDmPMlL8iY0w/MNtbu3qgfdLXHeOPXSYitwJrjTE/nulCIvIJEdkvIvv7+/tnua2ar3AijNs29/0I
Zk0Ee/8Rmt+S2XJyKTm9sO5OOPrYlEMHew/SVLoIS1tksalsE/t79ufl2krlw2yJID7PY7MSEQv4a+DTs5U1xnz
NGLPLGLOrqqpqbdVM5j3EhMz9RFEx+Dw/80s/1AIG98MB781ZcP7A70H2FS6KS+33FS2idd6tKVTXT9mSw
Q3ichYlo8AsGOWczuByVNHG8Yfm1AEbAdeEJFW4C7gCe0wLpz57FDmss+y1tDRxzJ7D/sLNBG9ZjvEAtB9+
PJDiXSCU0On2Fi6MS+33FK2hYO9B7WfQF03ZkwExhibMaY4y0eRMWa2pqF9wCYRaRQRJ/BB4PLS1caYU
WNMpTFmgzFmA7AXeJcxRuvUBRJOzq9paNoagTGZxeU2v30RopsnsTJ9BYf/7+WHzgydocpThdeRn0ltVZ4
qDIaOQEderq/UYlvcnrJJjDFJEfkksBuwAd8wxpwQkc8D+40/xup/BMhNNRufcWeywHCTTSZLp5NSO10uvQD
IGtbNVHucvkkhxrGOEUz0BukaiBKMJ0sbgtNvwOW2U+Zw0urbxxlOPEL7vjykt8nO473DeagOQmU+wuWw
zB/oOsLY4D+spKbXI8pYIAIwxT5KZhTz5saxzEIwxD+QzFjWzRCpB2qSxy9x+JUTk8r7FRc5rNpfZ9/XMEM55b
vgyk+FwnB8c6mRPywD1JR7WVXjZUuPH67QjAsm0IRJPEYwlOT7moylVzl9/8W/oqHkzjrqXuXPt5kWPabKN
pRvZ37Of9zS/J6/3UWox5DURqOtHOBnGbXfPa5cutz0zl+CqRBAZhnPPwHu/Ov2J85A2hp8c7+Hxg53saCjh
E/c1UeyefW6Cu/hNfC5whMeaPsQ/XTrB+UsbsN3ez70bK+e9M9lMNpVu4hvHv7Ho11UqHxZxMXh1PYskI
3js8xvemXUI6dHvQMOuzM5hi2QkkuALPz7FC2f6+eW71/PQ1pqckgDAaM2dlPW8QnP5MC5Hkvds38b3D
nTwyNNnCcaSixbjhIaiBgajgwxFZ174TqnlQBOBAuY3h2CC254lERz81qIuJ3GmN8BnHj9Khc/JL929nkr/3Poy
0g4/wYqd9F38N9b4GllX7uVX72nEZsEf/uAYvaOLO5/REovm0mYO9R1a1OsqlQ+aCBQwv1nFE1y2a1Yg7T
memc1bt3NRYnvmZC9f3H2Gt22v5YEt1VjzbMoZqbuHSwMvUetZB4DdEt62rZZb1pXy2SdO0Dk8j72XZ9BU
0sTBXt3RVS1/mggUsPAawVVDSA//KzQ9sOBtKOOpNF998QL/cbSLX757PZuri2Y/aQbBips4QYgG6+rr3L6+n
Ae2VPH5H51a1JpBc2kzB3oPLNr1lMoXTQQKmN+CcxOuqhGkknDsu9D0pgXF0x+M8cdPHKc3EOGj9zRS4Zt
fbJMlBc45HGwdujjl2M76Eu5truALT54kEEks+F6QqRG0jLQQS8UW5XpK5YsmAgXMb1OaCU6b80ofwYUX
wFcJJQ3zjuVQ2wh/8INjNFf5+YVbGnDZF+fXtDXWTYW9mJqeV8i20+pt68rYVF3El545Qyo1/T4GuXLZXTT4G
zgxcGLB11IqnzQRKGARawRHH4UNb5zXdVLG8Ni+dv7Xiy38ws313NW0uEM7z0YvscZVj7FseIezb27/pi3VJ
NKGx/a3Zz0+VxtLN2qHsVr2NBEoIFMjmOt+xROcNiehRAjiYTi7Gza8Ye73jyf5q5+c4VD7MB97QyPrKnzzimU
mZyOXWOOsIlS+g9LO57OWsQTefVM9L57t53jn6ILv2VTapP0EatnTRKCATI1gvonAbXMTTATh7FNQuTmzZ
/AcDEfi/PETJ7Fb8OE71+F35WfzmrPRNhqc1QTLt1Pcvx9JZe8Y9jltvGNHHV95voVQfGFzDJpLmznaf1QXoFP

LmiYCBUAoEVrQqKFQIpSZRDbH2sBwOM7nnjjBxiofb99eh22BI42mM5QcI2ZilNtKSDn8RP3rKOmdfqnojVV
+mqp8/MurlxZ033J3OU6bk0tjC7uOUvmkiUABEIwHFzSzOBAZhtaXYd1dc7hnki/8+BTbaku4b1NVXpZ6mH
AueokGZ83lewQrdlLW8dyM57z5hmoOtg1zqntsQfduLm3mcP/hBV1DqXzSRKAACCQCc96LYILH7iE00gq12
8Hpz+mcVNrw10+fpb7Uw73NFfO671xM9A9MCJc04wz34gx1TXuO227joa01fG3PBZILGEXUWNKoE8vUsq
aJQAHjTUPzTARuu5tQoAPW3ZvzOY/uayOWTPPQtpq81gQmnI62Uu+YtDmO2AhWbJ+1VnBDXTE+p42fH
O+e972bS5s53Hd43ucrlW+aCBQw9z6CRNhw7rEI+/8sQOzxKpIBJ6y9M6dzj3SM8NK5Ad5905p5LxcxF/F0g
vZ479WJAAhW3Exp9x4kPf0EMgEe2lrLDw93MRye3+6sDUUN9IR7GI0tfBSSUvmgiUABmUSQax9BMmI49X
/CiBiafsGN093PLWc/TSg++/nBWJJ/fOE8P79zDT7X0qyCfjHWRZW9DKd19WikhLucuLea4hk6jQFKh528M7
6WJ3/UO+von+hgmq6XYwydTGDSmbJ2y05TSRPHBo4t7BtRKk80EShgatNQIpS+/EI2mUkZzn0ngqdSqL7di
d0tlHr20F16lqd+kCSd5RyArvY0P/lhkm/+c4AtRRU0Vi7ePAGThv42i86zNuJZRoSejV6i3pl9z+RA5a2Utz897b
XjByC+R6j3uqk+6+XIo8GszwvAwLE4J74eItSZouP5OOcejZBOZso2ljTqxDK1bOnGNArITCjz2D2kE5kmn8ClF
Da3sP5hF+U3Zt5JG2NofTJKOm6ofsP4nINkFOdIG2erRrl7+A5efTHFvW+6+tfq0GspDr2epmxdnMFYlNoLax
hcE6eiPksHrDEQDBNuH2Ow3YYrOUaldGCZFNht4HaD3wfFfij2k/IWcXxvMbGI4PIaWg872PFgjOKKKy/Wpy
IXWe+qy/59l2yivP1pXIE2YkXrrjqWPA/JM4LzboO4hFBxlPRJw8Uf2Wn8+as38Rk+naDtqThrH3LhLrMwaUP
nC3Fan4zS9C4PzaXN7OncM58fjVJ5p4lAAVc2rm9/NkY6CZs+6CYykKbt6RgDRxKU3+hg6HiC2Kih4UEnYo2/
CPadxlFUS8IEuelO4WfPGErLUtx4s434wCgvPxmirdfL1the+n/SxRsdQsJdxaln38jNYz+kyAxBOg3RGASCMDJ
GZ/XdXFj/DkrjHUQddbRLDTvjT+NMhSCRhEgUwhFMKMKJtR9G6GHT0MuIx81I6UaO/fgWdvlewFXmwJQU
YQLn2VK3HodnDGwW6ZRFfMiPJQlcvlGiZhu1+79P75YPYux2Ui4XiZSL2M9cOG83yPjKGxtq/fy0v5eScw7sz
wpr3+JCRBg5l+TCv0dpeHMmCQCIJdS9wcmlJ2MMHE2wcetGvnr0q9n3dlaqwPQ3UpFMJ0mkEhCyM3A4T
OO73IgleKttbPh5i5GzKQYOJ3CVC9V3OLDskzp4e45ilW7AETyFsSe47YYxXnvGw9F/7yNqPDjtKbY7X6fHZhit
X0N5TTEOY6iPtnPM9m5u872I054AhwO8HjpijbQN1tHc0IPLaceYMXoHyzgYej+33HARp+PKTN+WtlriAR9N
tR0QuwsTi1MSjRELd3I8uItbux4nERzmI4Eo1amXsBIpRryNnGn8MM5kgKTNTSRVQW37XvzBi5S5voKkDMR
THN3y69SFuqk4dZpEsZ9YZTmRNTXcU1nFTyNdvPPMWoJtKZzFFmMXU9Q/4MRTeXVLq80prHmjk0tPRdla
4eSeDi9n//KPKeocIxkKMVbUiHfrFjb8pwdxVOR/CK1S05Hrber7rl27zP79+wsdxooyGhvlbY+/jc9YjxDqSVF7
Zw5LTSSTyMgI/PQr2Ip30XLuJNv6HVg2O/b1awlWN+OtKKK0OMFYNM5PjvVy18aKq1YS7RkoJRj2sHNzK05
HkkvdVXT1VbBxbfdVL/gTZceCPrZvuoTHFaO1q4a+wVI2ruvCbru6ickYaO2qweeOEq5+nhOR87yh6GaCI7U
MdW6luOoCLu8YxkBkrIbQSB3rix7DWR5jpPY+xs7XkAy58K/pwR6L4AiGsY8GcA2N4OrrJ54ydK9bR8mOO0
mWV+FvdGB3Zxn9FE9gnbnI2MFh+uLrWdfzTbzr3Pjq386x/lowaZLxNGUDJ7jvrWVUfvzXEEu77VR+iMgBY8
yubMe0RqAub0ozcChB7b1OiESxzl7Cau/G6htERgMQiUEsjiQSEE+CMRi3AxxuTPUAUa8NeWgXFVWZtvgSA
BKAYe/5IRqrfFOWk66pGEHE8PqxzViWweVM0Ly2C4cjNSXG2soRnPYUh041kTaC3xOlae3UJAAgAutq+zjfU
cdIbCvVpTaGujYTGqmjtO4MDmfkcjlvSS92Z5i2vv8HX+AC4Z51IIJ/XT/YbCSdfpJFfqirJgBgQIZHGT16noY9T+
EaHsHU15Cqr4GSosx409EgVkcPVmcvpqaS0sZ6THmCFu9/waRslIXsbNlhp7LKRjJpeP3V2zj9/LM0nvo09Y98
EbHrn6VaWvobpwgmglTH1pKOpyj68RPYT53H1FWRqqogXVsFG9dj3E5w2jF2O9jtmY7b889mZhKXrqNz4
DA7i6bWJFr6gsSSadaVeaccE4GailEqSgOkUhZOR5KZphWUlwYoLQmQTltZE8BkNpthY0MPL3emKe99kIQ7
Qtmak9jsUxeRc3oClK89jtU7jN2bIF1TNn0cAqa8hOQt2/jhUJif31KJs38QGRyBrr5MGY+L9I3NJB+8G1yZ56Q
Eg1R28GLby/zmrZ+4fDm7Xdi23c0x81bWdHyVni/8GXWf++MZvzelFpsmAkUoEeK2o+sp6TwOdU7iv/Tuyy
9g00olYbQTGu8HwCk24umr38lHEikOXBrh1vWlmXfK07Db0rO+sE+wBKwcy8YlTLdvD1uKIjPeH8Cy0jjKUpS
MvkA/72G2E2pK3PQEYhzpC3Hb+jWwbs2s8RS5/KQlxmhslBJXyeXHS8ss3G6L8Dt/g/T/+VPGnnmG4oceyu
E7VGpxaIOkIrH7OWq71+PdXkH6tu2zJwGAsXZwl4A9U9YuNmLpyTNvDXvPD1Jf6qYoT8tKz6Y3MUSZrWjW
JDAh4S7HYMMdbMup/Na6Ilr6gvSNRWYvDAhClaeKzmDHlGP1DTbOX7Io/9Vfpffzf0o6FMotaKUWgSaCVS
52/jyuv3+MQNE6PA1z2KFs8AL4ay5/6bDsxMyVZpfzfUFGI3GaqnJbhC4fehIDlNiL53ROtLgR/8CRnMo6bRb
b6kp46dwAseTUFo1sKjyVtI9N3f2sutqitzcF6zfjbG5m8FvfmlPcSi2EJoJVzBhD9x/+EV33PkjKPYbNnuMIsnQS
RtuhqPbyQw6xE0tl1uwZiyTY3zrC9vqSJVlLaDo9icFMjWAO4p4qBIM7y7v2bCqLnFT6XfysZRCTZR/ka1V5q2
gLTk0EdrtQUWnR0ZGk5D3vYehb/0w6HJ5T7ErNlyaCVSz4/POkRobprW4GzxwWRBvtBHcR2K/UIOyWjWg6

TjJteOFMPxur/XnbaSwXMRMjko5SbJt7jSRS1Ih/4HDO5TfVFBGIJjjVNfu+BWXuUkKJEJHk1Bf5qkqLtktJHLW
1uDdvZuQHP5xD1ErNnyaCVcoYQ//ff5nid72byEgx4gnkfvJgC/ivXrLBIXai6Th7zvbjc9tpKJ3fJjeLpSc+SJmteM
ZRSNOJe6sRk8Kd5Z17NpbAjvoSjnWMztpfIFhUeSrpCEytcVRW2ejuTpFOG/xvehMjjz029+CVmgdNBKtU5N
BhUiPDuG66mVSgGIcvtw7PzGihdii6OhE4LRtdYyFC8SRba3PvoM2X7sQgpXPsH5gsUtxEUX/ui8R5nDa21Zf
w4tkBwrPsc1zpqaJtbGqHtMsleL1Cf18a19atpEZHiJ48OefYlZorTQSr1PCjj+K//35GRsHYozhdOb5yj7aDu/jya
KHL1wumCMSj3NRQWtB+gQk9iQHK59g/MFnck9nNzDPWmvM5lX4na0rdvHSmn3R6+iGu1d4qLgWyj0wq
r7DR0ZlELAvvHXcy+qMfzylupeZDE8EqlI5ECD73HL6776G/P03SNThlrf5pDbZMqQ1cGgozOJbA4xEctsL/SkX
SUaLpGEXz6B+46jolzfgHDiEm920qm6r8pDEcbBuZtky5u5zR2CjR5NQ1sysqhK7OzAgk7x13MPbkk7PugaD
UQhX+r1YtueCLL+FsasJWWkpvb4qoYwCnLYe5A6k4jHaA/8poofahCBf7Q2woLyLFzE0iS6U7MUi5rWRe/Q
OTxd3lpG0uPCMtczrvxvoSLg6EaBvKPhfAEhtV3iras8xXKCuzGB1NE4sZHGvXAhA7c2buwSs1B5oIVqGxp3fj
ueUWAPr7UoQdPdhzqREMtYK34nKzUMdIhJa+IJtrivA4nMRn2PJxKXXH+ymzz79ZaLJwSTP+waNIOvck57R
Z7Kgv4dWWQYKx7M9JlaeK1tFLUx63LKGszKKnO4WI4LlpJ4Hnnp93/ErlQhPBKmMSCUJ7XsZ7661Eo4Zo1B
CSfpxWDjWCwXOX5w50j0Q41xNkc60fl8PCjo2EWQY1ApNJBOWOktnL5iDlLCbpLsU3eGJO55V4Hayv9PLS
2ez9BTXeatrGpiYCgPIKi86uzHPp3nkTwec1Eaj80kSwykSOHMFeXYWttJT+vhSlpRbxdBTHbE1D8RCE+qGolr
5AjFM9QTbV+nE7bEBm+GjCJMlhTlVejaaDGMAnizd8NVLcjG/kNLYsY/9nsr4isx3nkY6RKcfK3OUEEkFCyanN
RxUV1uV+AvfmzcRaWkgF5jC8V6k5ymsiEJGHReSMiLSIyGeyHP8dETkpIkdF5Kcisj6f8SgI7tmDe9uNAPT1pS
guERKpBE7bLOsPDraAv4bBSIrjnWNsqvbjGU8CkFlJ1CYWiQL3E3TH+6lwlCzq8NWU3U3MX09R/8E5n3vjm
hLO9k5dj8gSixpvDZeyNA/5/UIiYRgbSyNOJ65Nmwi/9tq841dqNnlLBCJiA74CvB3YBnxIRLZdU+wQsMsYsxP
4HvBX+YpHZQT3vIz7xkwi6O1N4StOYbPsyGy/CgNnCXlqOdI+SlOVD6/LNqWIHXvB+wk6En1U2BenWWiySF
EjrlA3zsjAnM5z2jPrEe05N0A8dfV6RDXealpHL045R0SorLTR2THePHTDDYReeXX+wSs1i3zWCO4AWowxF4
wxceBR4N2TCxhjnjfGTNS39wINeYxn1UsFAsQvXMC1cSOplGFwMI3bH599xFCon3Qiyr4eWFfupcidvfbgEA
dxU7hEkDJp+hNDlNkWPxEYy0a4ZCPFva/P+dzKIidlPievXxi66vFaXx0Xp5mnUFll0d6eSRyuG24g9Prc76tUrv
KZCOqByXP0O8Yfm85/Bp7KdkBEPiEi+0Vkf39//yKGuLqEDxzA1dyMOJ0MDqbx+YSURGftKE71naY9XU51iY
cy3/Sji5yWjZiJT3s83/qTQ/htXpx52hw+5lsDJolndG7DSQE21xTRMxbl0qQhpX6nH0ss+rPUMioqLPr6UiQSB
uf69SS6ukgODy8ofqWmsyw6i0XkI8Au4IvZjhtjvmaM2WWM2VVVVbW0wa0g4ddex7V5EwC9PUnKyiyiqdi
MNYJ0KkG6v4WQq46a4pmXqS5001BHrJfyPNQGJguVbaW4/yC21NwSns0Stq8pYe/5QSLjS1AIQp2vlosjF6
aUdzgyw0g7O1KI3Y5rUzORQ4cX41tQaop8JoJOYO2krxvGH7uKiLwF+EPgXcaYWB7jWfXC+/fj2rQZgO7uNG
VlFpFkBNcMieD8meOELC81lWWzXt9p2YkWsEbQkeij0l6a13uknEXEPDX4+/fP+dwSr4P6Ug+vnL+yZHWNr
5bzo+ezlq+qtrhwMZNYXU1NhA8emH/gSs0gn4lgH7BJRBpFxAl8EHhicgERuQX4Kpkk0JfHWFa9dDRK7Nw5
nBs3kk4b+vpSlJVnEsF0NYIzPWOUBVugNLfBXHaxEytQjSCYihBNxylZ4LISuYiUNOMKduEM98753MYqP2OR
BOf7ggDUemrpCfUQS019D1RTY6O7K0U8bnBt2kxk39yTj1K5yFsiMMYkgU8Cu4FTwHeMMSdE5PMi8q7xY
l8E/MB3ReSwiDwxzeXUAkWPH8fR0IDlctHfn8brE5xOIZKM4MjSR9A3FuF8WwelVpiUtzqneziwX7Nd5dLpT
PRSZS9dklVPjWUjXLaFkp5XkHRuO5NNsAS215dwoHWYYCyJ3WanyltF62jrlLJOp1BeYXHxYhJnUxPRM2cw8
cLVuNTKldc+AmPMk8aYzcaYjcaYPxt/7LPGmCfGP3+LMabGGHPz+Me7Zr6imq/IkSO4mpoA6OxIUlGRGf4Z
SUZwXbOSaDie5IWzA9zh6yHuX0Oui/Y4xF6wzuL2WA+VjtmbrxZL3FNFyllE0cDc5xb43XbWlXv5WcsAxhjqf
HWcGzmXtWxDvY1TpxKI2429pobombMLDV2pKZZFZ7HKv/CBgzjHE0F7e4rKisyLeyQRuWrUUDqd5oUzfa
wrcVAeaSPuy31Eb2bf4qVPBEmTpD85TEWeO4qvFS7dgnusFWe4Z87nbqj0EUukONMzRoO/ngujF0hnWe
W0otIiETf09qZxNTUROZLbfspKzYUmglXAGJOpETQ3EwymCYXSlJZlfvSRZBiX7cpooInlk29w9JBwlZOyu3O+T
6Gahrri/ZTYirBbUye55VPachAu20ZJ98+w5jiKCMmsUnq4fZRE0oHf7qMjyz7JIsL6DTaOHonjbGzUDmOVF5
oIVoFkby8mmcRWWUlbW5KqahuWNV4jSEZwjieCjuEQFwZC3Fhfgn/kNDH/2pkuO4XDchArwISytnh3pn+g
AOKeChLuSop7XpnzuV6njaYqHy+3DLDGX8+ZoezLTdfX2xgeTjNWuZnosWMLDVmpKTQRrAKRo0czE8lEuH
A+SU31lR97JBXFaXMRiiX5WcsgO+qLKYp0A5Bwl87pPg6xE1/iGkHapOmI91PlKF/S+04WKdmEIz6Kd3ju+w
asLfeCgUi4lLNDZ8m2ap9lCY1Ndo62lZAcGCQ1MrLwoJWaRBPBKhA5chTn+vUEAmnGAmkqKq/82KPJTB/B
nnP9rC3zUup14hs6SdS/bs73cYqd+BKvQNqbHMJrc+POZRntPDGWRbB8J0UDh3FEB+d8/ra6Ylp6UtgsR9Z

N7QHq6y3GxgzBzfcSOT63JbGVmo0mglUgcuQwzsZGzp5NUFd3pVkIIJaKc7Y7TDJtaKz04YgNY4+PEPfWznD
F7EQEm9iIs3TNQ5di3VQXqFlospTDQ6hsK6WdL8551rHbaWNTtZ9ouJQTg9k3q7csoWmjnXMV9xM+enQx
QlbqMk0EK5xJp4mdOo19QxPnziZpaLjSoRpNRrGwcaYnxPY1maWbfYMniPnXYua5z6NLHETSU/fizQtjaIt1U
13AZqHJ4t5qEp5qSrpehDnuM7ymzIObGk4MnM46egigrs4iavnpOjplgr5SC6KJYIWLX7yIVVRE+4ALr0/w+6/
8yEeiAVIpO1vrinE5LGzxIK5QJ9E5dhJP5rIcRNJLs1JIT3IQl+XAay3eJjQLFS5pRtIJiuexd8H2uhqSSRcHurL3NVi
WsG6tcDow/5+PUtloIljhIseO4Wxs5MSJBOvWXT288tnT7TgsN1Xji8n5h04Q89VjFjAM0ylOokvUYdwa66Jm
mdQGLhMIVezAFWjFk2WvgZk47RY1nnr2XDpAMsv2lgANG/2MeuoZPNWe9bhS86GJYIWLHD1KoG47kbC
hetJooWOdowyERyl1ewGwJSO4A61Ei+beSTyZU+xL0jSUNmkuxbqpcVTm/V5zlbYcBCtvorjv9TlvZLOptJG0v
Y+nT2TvNLbZhep0F0ef0hnGavFoIljhokeOcp4trFtvR8bb/YfDcXYf76Gp2oHTytQGfEPHiXlrSc+2Sc0sHLI0TU
PdiX68Njcea+alsQsl5fATKt9GaecL2OJT9yWejt3mpMJdw8mhU1wcyH5eTUmE8+eTpFPZaw1KzZUmghXM
xOOMtA/RG/BQX5/5UafShscPdLK9vgSxRXHYXNhSETyjF4gWbVjwPV1LlAhaou3U2pdfbWCyuKeKaNE6yjt/
ipXKfSTVGu96PEVt/PuhTiKJqYvaldaX40wEaT+tG9WoxaGJYAWLnj1HV9ND1NfbsNsztYHnTvdhtwk31BYRS
YVxWi58g+O1AfvC3107rfyPGkqYBF2JAWodFXm9z2KIFq0j4SymtOsFZJrRQNcqc1WRljhVZRH+40jXlOP2ulp
KBk5xbt/cl8FWKhtNBCtY6MhROktvpmFtpvO3pS/I8Y5R7m6sQASiyRBuI5naQHHjotzTKQ4ied5f6EKskwp7
KY48bUm52MKlN4BJUZLrMhQi1HnWY/e2MhCIs6/16r2OLZ+P8mgHFw/1kUpq85BaOE0EK1jL/l78zgQ+n8
VYJMG/H+7i3uZK3M7Mjz2ailASuETMt2bBfQMT3JaLSJZNVhbT2Wgb9cuwk3haAqHyHdhjIxT35rYJfZ1nA1
3hi9y50c+LZ/vpGI5cddxbW4rXmaTr7EgeAlarjSaCFezCcBn1a4RU2vC9Ax1srfVTU3Kl+SeaDFIS7CZavGHR7u
kSB3GTJJVjM8hcDSVHiKZilC+D2cRzYSwbgcqbcYV7KO7bN2t5h81JpaeOvvgZ7mqs4Lv72hkJXxmW66ipoTT
VR8sh3dhPLZwmghVquLWfgL2CuuZSdp/owWYJW9dcWa8/bdLEUxHwryNtORbvxgIey0k4HZm97Dycilyiw
VWd6145y4qx7IxV3YIr1E1Jz6uz9hk0eJs4N3aE+jIX2+qL+fbeS4xFMp3O9jVrKOo+TuvRzOY2Si2EJoIV6uSTJ6
hIdXO0e4yWviD3NFcwaYkhUmPn8BiI+3Pbj3guPJaLUB4SQczEaYt3U+/IbevM5chYDsaqbsMeH6Ws/afYUt
N3rPscJfjsxbQGznBDbRHNVUV842cX6RgO46itxd55FhEY7AwRS6Z47nQvf/nUaT73xHF+crybVFoThMqNJoI
VyBhDy+kIPleQZ0/1cf+Wapy2ST9qA7auF/HYPBhr8X8FnJaTYB4SwdlIG9WOcpyLWYMpAGPZCFTcTNrhoaL
1R7izbEgzYa2vmdMjB0gbw7Y1RexaX86j+9r5zqEeov5SklaYr/zLUW7/wrN86emz9AdjxJKGLz19ll/8+muMh
pd+fwh1/bk+hl2oORnsDBKPJjkUinLPLRWUeq7+MfsGjtApKZz2/Gzt6BYnodTiJoKUSXMqcoFbfDcs6nULRjLr
EiVc5RT1vo535AyBiptIeK7uBC91VmFZdjpC51nnb2ZtuYfaknpaB0L0+8pJDLVRkt7Cn39oBxX+K/0/79xRxzdf
beXX/+UA3/7Pd+Cw6Xs+NT397ViBDu/poHjgNJWbNlBfdvWCbJKKUdb5Uwb89bjytIa/W5wEU+FFvebFeCd
FNh9FNu+iXrfQEu5yxmrvJuksoazzBSouPYlntOXK1pcCG3ybOT78GunxvgCHTdhU46dm0wa2Ri7iCqUosV2
9PpRlCR+9ewOReJKvvnh+qb8tdZ3RRLDCjITjHN3TjjfWxca1U4dYlrc/S9zXQMACt+RneQav5SaQyn1Zhdmk
TZojobNscNYt2jWXEyNC1N/ASN0biPnX4hm9QNWFxylvewbf8CkqxY1NbLSFrl5fKF27BnvbeTzVbsZaA1Ou
a1nCx+5t5J/2XKR9aHETs1pZNBGsIKORBL/55b240imKqqe+c3YF2/GMnCFQeQuBVCRvu3p5bR7GUsFFu96
FeCcey0mZo3jRrrksSWZZimDlTkbq3kjcV4sz1E1F+zPcPNbPyb7ncQ2dRMbnaRifD5xOfN4Eoy1jWS9ZXezm
LVur+ZtndJE6NT3tI1ghBoIxPvL117g7ZlGV6IC6+qsLpFNUXPwRgarbSNucjKWCbHQ35CUWtzhJkiJuEjhlYR2
7KZPmcPAMO7zNixTd9cFYNuKeKuKeKgA8iSC+cAsX+l/m7os/Iu6tJlyymVhVBcXBS1wcXotJG8SaOq727dvr
+PR3j3C+P8jGKv9SfyvqOqA1ghWgpS/Ie77yM7avKaZuLE1Z+0HS1ySCso7nSDmLiPo3kDJpIuk4HnHnJyABv
+VldBFqBScjFyiyeSmxr+4XsJTDz8aiLbzuhEtN7yRcuhlXuIvi9Dkq9z+GR0bg+GuQTk451+ey89ZtNfzjC9pXoLL
TRHCd+8nxbt7/v17hnTvreHhNBZJM4nbEMd4rTUOusUv4ho4xVn07CARSQXyWO6+Tsrw2z4ITQSQd5UTk
PJs8C9sjYaXwWR7qnVXsD58h5qtnrPp2xrY/TGzUQZmvn8Tre9n13dtpeuX3KOl+GdJXVi59aFsNu0/00Du2R
NuIquuKJoLrVDCW5PceP8rnnjjJpx/awv2bqxk6NUyJ6Sddf2UrQysRoerCDxmruZu0LVMDGE2F8Fp5qg2M8
1sehhPZ261z9XrwOGudNXmP9XrS5KynOz5Id2J8wxuPG+P34/K7OZ+8n4t3fB5jc7Jh3+e57fF7WHfgf+Aeu0
CR28G9zZV885XWgsavlidNBNcZYww/Od7Dg196gf5AlD9773aaq/2YtGH41AhlXYdIN6yfKEzlhceJFK0j5ltz+
RqjqQBeW35fXIstH/3J+a+X3xbrZiA5wgbXmtkLryI2y8ZWTyOvBA6TMOPNQLVV+LrOkUwII/FqBte/g4t3fJ5
Lt/wuznAP2596P9t2f4iPlh3nsb0XCcenNh+p1U0TwXXkeOcoH/zaXv78yVN8/L4mPn7fRrzOTH9/oC2I3QHe
3gukazPDLEs7nseWihGsvPmq6wwmRymy+fIaa4nNz0hqjPQ8Fp8Lp6PsDR5lu6cZm+iv6LUqHaWU24t5PXg

MAFNXg7RcpKQ6Rf+lK/MJ4v56+jZ9iHP3/R2B6l3sOPdlnrJ9iqPffwQS+VkLSl2f9K/sOtA+FOa3/u0gv/yN17m
xvpg/f+8Oblxz9azgwWODlFoDpNZtAJsNX/8R/EPHGKl7A1zzYjqYGKHYlt/OV5tlwyMeRlJTx7fPJG3SvDC2n3
Wu2lXfQTyTza719CQGuRDrhIoyiEQp9YzRe9HGtWvQGcvOaN29tO76Yy42/wqesz/E/M12eOkRiIwUJH61v
GgiWMZGwwk+/x8neMff7cFls3jk/Tfx0NZabNcMEUxGkoxeCFB+8VVS6zfiGW2hrPNZhurfdLlfYEIkHSVJCm
+eJpNNVmr30x2fw+btBl4NHsWGxQZn/ezlVzGbZeMm72ZeDx5nKDUGDXV420+SSgjB4WlGAYhQtO4m/tX
zSxy94VNw6RX4nzvhyd+FgZYljV8tL5oIlqFU2vAve1t54JHnaR8K85fv28n7bluLx2nLWn7g2BD+ShuO3nYcZU
LFhX9ntO4BUs6pawkNJEYosflhCZZxrrKXcSnenVthAwfDpxhMjrDds3FJ4rve+W1etnoaeW70dcINVVgnzlJW
l6L73MzTg+7eWMG3zwjpez8F7/xbiAzB/34Ivv4WeO2rMHSBKdUKtaLphLJl5njnKL/3+FGMgd97+AbWV8zc
lm9Shv6DAzSYM8jaGiovPs5o3X3EPdl38OpM9FNmX5oZuuX2Eo5FWoiko3hmGvljDPtCJ+hM9HOrdys2K3v
CU1PVOMqJpeP8JHaC9yKUSzfnWhtoujWBfZq5fJuq/ew518/+1iHuaKyCW38FbvowdB2Elmfgxb8CywZ1N0
H1NqjcDBUbobwJfFVcl5tBqBlpIlgmIvEUX3r6DI8f7OADt6/ljZuqkBz+4IZOD2N3W5Tu2U1pc5DRujcS99ZkL2
ygM97LTd4tixx9dpYIdY4KTkUucqtva9YyCZPk5cAhgukIt3m34dAkMGfrXLWkSXO4rosbD76Kf/P76G6xsXZr
Kmt5EeH+zVX862tt3Lq+HLslYHPA2jszH8ZAsBcGW2CsA44fh0AXjHVmjlVuhrqdmURRuwOqtoJzZS0GuNpo
IlgG9pzr5/e/f4zGSh9/8Qs7KfbktixDOpWm6+UemiIv4fRECDS/iYSrfNryw6kxjDH4Lc+0ZRbbetcaXgseZ4t7A
z7b1fftTQzys8BhyuzF3ObdiqXvNOdtg2sNPRssks9fJLHzNO0ntrFmUwrbNH/hG6v87Gsd5qnj3fz8zmuG6Ip
AUW3m41rRURi+BMMX4fSP4JUvw0gb+KsyNYaSdVBcB/4a8JaDtzJTiyiqBU+Z1iaWKU0EBdQ7FuVPf3SSfa1
DfPSeDdy8tmxO5/e90k5J8hK+I68Rf+PdpGdIAgBnopeoc1Ytafu7x3KxwbWGnwZe5w3+Wyix+elPDnEqcpH
+xDBbPRuocswct8pNrbeWwOYInlefo/fGMs4eK2brLdlrWCLC226s5ZuvtHLr+jLqS3J8c+AuGa8N7LzyWDoF
gZ5MrSE0AKOd0HcKYmMQC0BkOPO4SUPpeqjclGlyqrkx81HWCHnYIEnlThNBAYxFE3z9pQt885VWHtxaw
1/8wk7cjjk2iRx9hcHXDVv6nyfd2ISpmKY5aFw0HaM11sld/p0zlsuHDc46bAjPjb5G1CQosnmpc1Sy2b8em7
4ALKr0xg1Ud4wxEH6dttNv5XDRHh5cfyPVWZJtuc/JA5ureGT3Gb7wnu34nPN8ObBsUFKf+ZhJPJhJGKMdm
Q7piy/C0MVMwqjaAjXbM4mh6obM10V1WoNYIpLPja9F5GHgfwI24OvGmL+45rgL+GfgNmAQ+IAxpnWm
a+7atcvs378/PwHnWUtfkEf3tfHd/R3cvLaU991aT1XR3Gb4ekZbKPnZ13n+1DtYkzpBUcdZzJvuhVl2oHo5cAi
DYZNb1+1Z6WzBCJUvHef8PQ8wHNrAj278MpWlbu7138xtvq2U2ouuKr/7RA+DwRifeftWitwFeG8YD8JQK
4y0ZpLEaAeMtmcmvZU3ZTqqKzZBRfN4p/XGTLOTJok5EZEDxphdWY/lKxGIiA04CzwEdAD7gA8ZY05OKvOb
wE5jzG+IyAeB9xxpjPjDTdZd7IjDGEE2kGYnE6RmNcqE/xJGOEV4+N8BIJMG9Gyt4y9YaqotzTwBWIkhp10tUn
nuc9tYSXh/7AGsCRykbPIW5705wzb6vwPeHfspNvs34lrB/QBWOcyhI2Wun6dz6BobsWxnddoSjZa9wPtZB
haOU7Z6NbHavp8lVT5W9jJfODnC8a4xfvnsDdzaVY1sOL7LxYKaZaawLAp0Q7Mt8PtqRSQKl6zMfZRugdG2
mBuGvyfRJeMvBXapNTpMUKhHcDXzOGPO28a9/H8AY8z8mldk9XuZVEbEDPUCVmSGohSSCrpEIPz3dB8Z
gyIzXT6UNiZQhnkwTTaaIxFMEokkC0QSjkQQjkcz/gUiCUDz7KIyZ1JW4WV/hZWtdMRur/FOa5x2JABu6n8S
WjmNLxbCnIjiTY7hjA/jDHZQFznG04wEOm/cRdKzBloxS3/USTneMZH0VZFl/PpvDqTaqxI9DWwNXDXsiSXlb
gKRVRUf9/SAWrlQr3Rt3c3jNID1mlDGurEbqN25sMSfOlIMiy0Wxw8X6Ej8lbid2sbAhgGAJCIKFZP6Xic/BJtak
x8iUR5Dxc678tmY+a3SVcbtvLXNiTKbvIdADwR4I9UN4AMKDEB6GdGKaJ8QFDl9mhJPDBw432N1gc2WO2
V2Z0VO2if+dmf8te+Z/sWWawcQan60vk/rbBBh/2TLj/5j0pI9rXtImrmFNXHPS/xOPXy4r4/cS2PIOKJq5GXg6
MyWCfL4q1APtk77uAO6crowxJikio0AFcNV0VBH5BPCJ8S+DIjJ4bZlc2Ioq62y+0oWtYmZM2pjxn2wmYZnJB
zGYyQ91AQdmuFyJFbGttw3O+JZ+JPIacftJhDTWNevNB2IBilxF05w5KbJl8AYv3wLRAEXu2Z+L1eCq58KA8C
+kLQcGQdLdjPqvvFimBZJy1S/ykrEMNIbjeVn4SAQZjhhHtU+wBBsEybRAX7+6g6ajK2B653n6+ukOXBdvD40
xXwO+NvG1iOyfLrOtNiKyfyA4oM8FmediUJ8LQJ+LCSKyv2Msveqfh9nkswGtE5hc52sYfyxrmfGmoRKu95Stl
FLXmXwmgn3AJhFpFBEn8EHgiWvKPAH8yvjn7weem6l/QCml1OLLW9PQeJv/J4HdZIaPfsMYc0JEPg/sN8Y8
Afxv4Nsi0gIMkUkWufja7EVWDX0urtDn4gp9LjL0echBXucRKKWUWv50kK1SSq1ymgiUUmqVu+4TgYh8Wk
SMiGRfgH8VEJEvishpETkqIj8QkdJCx7SpIi0i8pICx1MoIrJHROpEpOvBl8WRJ4XkZMickJEfrvQMRWaWaiNhE5JCI/Kj
QsSxn13UiEJG1wFuBtkLHUmDPANuNMTvJLOvx+wWOKe9ExAp8BbgXaIHsYSyLWWyal/8JojIBuAW4L

UCh1JIf0vmjWK6wHEse8t6iQkReRbIsk0Sfwj8AZlmoVVhpufCGPPv42X+kEzzwL8uZWxqeRERP/A48CljzFih
4ykEEXkn0GeMOSAiDxQ4nGVvWScCY8xbsj0uIjuARuDI+L6+DcBBEbnDGNOzhCEumemeiwki8lHgncCDq2x
2di5LmawaIuIgkwT+1Rjz/ULHU0D3Au8SkZ8D3ECxiPyLMeYjBY5rWVoRE8pEpBXYZYyZ84qkK8H4BkB/Ddx
vjOkvdDxLaXyNqrPAg2QSwD7gw8aYEwUNrAAk867oW8CQMeZTBQ5n2RivEfw3Y8w7CxzKsnXd9hGoq3w
ZKAKeEZHDIvK/Ch3QUhnvJJ9YyuQU8J3VmATG3Qv8EvDm8d+Dw+PviJWa0YqoESillJo/rREopdQqp4lAKaV
WOU0ESim1ymkiUEqpVU4TgVJKrXKaCNSKJiK1IvKoiJwXkQMi8qSIbJ7ntb4pIu8f//zEwu6icgfXFPuD8dX/z
w6PoTzzoV/J0rlz7KeWazUQoxPsPoB8C1jzAfHH7sJqCEzCQ0RsU9asC9nxphfm/TlHwB/Pn69u8nM8L7VGB
MbXx7ducDvY14xKpUrrRGolexNQMIYc3mC3fgihTYR2SMiTwAnx9es/6KI7Bt/F//rkEkkIvLl8b0OngWqJ64jIi
+IyC4R+QvAM/7O/1+BOmDAGBMbv9+AMaZr/JzbReQVETkiIq+LSJGIuEXk/4jIsfF18980XvajIvKEiDwH/FREf
CLyjfHzDo0vuqjUotAagVrJtgMHpjl2K5k9HC6KyCeAUWPM7SLiAn4mIk+TWb1zC5l9DmqAk8A3ppjp
fmkMeZmuLzg22dF5CzwLPCYMeZFEXECjwEfMMbsE5Fih+gJDmvXxjjNkhIjcAT09quroV2GmMGRKRPweeM8Z
8bHzjoddF5FljTGgxnii1umkiUKvV68aYi+OfvxXYOdH+D5QAm4A3Av9mjEkBXePvzmdkjAmKyG3AfWRqJI+
N75p2AOg2xuwbLzcGICJvAP5+/LHTInIJmEgEzxhjhibF+C4R+W/jX7uBdeimK2oRaCJQK9kJ4P3THJv8TlqA3z
LG7J5cYL7r9IwnjheAF0TkGPArTF8zmcm1Mb7PGHNmPjEpNRPtI1Ar2XOAa7zpBwAR2Unm3fpku4H/Mr6E
MyKyWUR8wEvAB8b7EOrIvMPPJjHp3C0ismnSsZuBS8AZoE5Ebh8vVzS+cuoe4Bcn7kvmXX62F/vdwG+Nd4
AjIrfk+BwoNSutEagVyxhjROS9wN+KyO8BBUaAV+OE1Rb8ObCCzp4UA/cB7yIw4ejOZvoE24NVpbvU14KiIHC
SzHPjfj7fjJ4EW4BPGmLiIfGD8mIdM/8BbgH8A/nG85pAEPjo+2ujae/wpmR23joqIBVwkMzpJqQXT1UeVUm
qV06YhpZRa5TQRKKXUKqeJQCmlVjlNBEoptcppIlBKqVVOE4FSSq1ymgiUUmqV+/8BmgJ4JfoAnXAAAAAAS
UVORK5CYII=\n",

```
     "text/plain": [
      "<Figure size 432x288 with 1 Axes>"
     ]
    },
    "metadata": {
     "needs_background": "light"
    },
    "output_type": "display_data"
   }
  ],
  "source": [
   "sns.kdeplot(df['CreditScore'], shade=True)\n",
   "sns.kdeplot(df['Age'], shade=True)\n",
   "sns.kdeplot(df['Balance'], shade=True)\n",
   "sns.kdeplot(df['EstimatedSalary'], shade=True)\n",
   "sns.kdeplot(df['Tenure'], shade=True)"
```

```
 ]
},
{
 "cell_type": "code",
 "execution_count": 17,
 "id": "7244c370",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:51:28.450803Z",
   "iopub.status.busy": "2021-12-21T16:51:28.449763Z",
   "iopub.status.idle": "2021-12-21T16:51:28.639792Z",
   "shell.execute_reply": "2021-12-21T16:51:28.639088Z",
   "shell.execute_reply.started": "2021-12-21T16:22:23.044838Z"
  },
  "papermill": {
   "duration": 0.228366,
   "end_time": "2021-12-21T16:51:28.639937",
   "exception": false,
   "start_time": "2021-12-21T16:51:28.411571",
   "status": "completed"
  },
  "tags": []
 },
 "outputs": [
  {
   "name": "stderr",
   "output_type": "stream",
   "text": [
      "/opt/conda/lib/python3.7/site-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"
     ]
    },
    {
     "data": {
      "text/plain": [
       "<AxesSubplot:xlabel='Exited', ylabel='count'>"
      ]
     },
     "execution_count": 17,
     "metadata": {},
     "output_type": "execute_result"
    },
    {
     "data": {
      "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAYsAAAEGCAYAAACUzrmNAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90
bGliIHZlcnNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXBIWXMAAAsTAAALEw
EAmpwYAAAUx0lEQVR4nO3df5Bd5X3f8fcHMP4dJGCjEkmumFglwWmNyQ6QupNJrUb8aGMxGUNx66lSt
cpMSBonrRvcdqoEwtSeuKXg1nTUICw8KT9MQlFTaqrKdj1pzY/FEMyPUDbYGGkAbZDA2NSkYr794z5rX8Su
zrW8Z3fFvl8zd+453/Occ56dkf3hY/w0KS1MmwkCR1MiwkSZ0MC0lSp2MWug
N9OPHEE2vNmjUL3Q1JOqLcd999f1ZVYzNte12GxZo1a5iYmFjobkjSESXJk7NtcxhKktTJsJAkdTIsJAkdTIsJEmdDAtJUi
fDQpLUqdewSPJrSR5O8lCSG5O8KcnJSe5OMpnk5iTHtrZvbvZNbfuaoeeN8tbfuaoeN8tNUfS3J2n32WJL1Wb2GRZCXwj
4DxqvoJ4GjgIuDjwFVV9U5gP7P7Cp7bIJ2N/qV7V2JDm17fcu4BzgU0mO7qvfkqTX6nsY6hjgzUmOAd4CPA28PA28D
7i1bd8OnN+WN7R12vZ1SdLqN1XVy1X1NWASOKPnfkuShvQWFlW1B/gPE8A0GIfECCcB/wfFUdaM12Ayvb8k
rgqbbvgdb+hOH6DPt8V5LNSSaSTExNc39HyRJS1hvv+BOspzBVcHJwPPPPAZxkMI/WiqrYCWWwHGx8d/4Dc6/
eRHbviB+6TXn/t+5+KF7oK0IPochvwobwNeqaqqq/h+BnCnPPCBtpDu/ewDngE+Le8BVgO07ccBzw3X
+gyLbwBnJXlLu/ewDngE+ALwgdZmI3B7W97R1mnbP1+Dd77uAC5qs6tMPaqyb7W7RsmAwvOhqMTUBNYKwPbcQMqvvlkPmKJ
nwFOADcz2CY6L8CNyX57Va7ru1yHfCZJPAPgYzoKiqh5PcwiBoDgCXVtUrffVbkvRavT5t2/TUcsOKj/BDLO
Zquo7wAWzHOdK4Mo576AkaST+gluS1MmwkCR1MiwkSZ0MC0lSJ8NCktTJsJAkdTIsJEmdDAtJUifDQpLUy
bCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0lSJ8NCktSpt7BIckqSB4Y+30THJ9mZ5PH2vby1
```

T5JrkkwmeTDJ6UPH2tjaP55k4+xnlST1obewqKrHquq0qjoN+EngJeA24DJgV1WtBXa1dYBzgbXtsxm4FiDJ8
QxezXomg9exbpkOGEnS/JivYah1wJ9W1ZPABmB7q28Hzm/LG4AbauAuYFmSk4CzgZ1Vta+q9gM7gXPmqd
+SJOYvLC4CbmzLK6rq6bb8DLCiLa8EnhraZ3erzVZ/lSSbk0wkmZiamprLvkvSktd7WCQ5Fng/8NmDt1VVATU
X56mqrVU1XlXjY2Njc3FISVIzH1cW5wJfqapn2/qzbXiJ9r231fcAq4f2W9Vqs9UlSfNkPsLig3xvCApgBzA9o2kj
cPtQ/eI2K+os4IU2XHUnsD7J8nZje32rSZLmyTF9HjzJW4GfBX5xqPwx4JYkm4AngQtb/Q7gPGCSwcypSwCqa
l+SK4B7W7vLq2pfn/2WJL1ar2FRVd8GTjio9hyD2VEHty3g0lmOsw3Y1kcfJUnd/AW3JKmTYSFJ6mRYSJI6GR
aSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhIUnqZFhIkjoZFpKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhS
SpU69hkWRZkluT/EmSR5P8VJLjk+xM8nj7Xt7aJsk1SSaTPJjk9KHjbGztH0+ycfYzSpL60PeVxdXA56rqx4B3A4
8ClwG7qmotsKutA5wLrG2fzcC1AEmOB7YAZwJnAFumA0aSND96C4skxwE/DVwHUFV/XlPAxuA7a3ZduD
8trwBuKEG7gKWJTkJOBvYWVX7qmo/sBM4p69+d0kbwVWVNXTrc0zwIq2vBJ4a
mj/3a02W/1VkmxOMpFkYmpqao7/FEla2voMi2OA04Frq+o9wLf53pATAFVVQM3Fyapqa1WNV9X42NjYX
BxSktT0GRa7gd1VdXdbv5VBeDzbhpdo33vb9j3A6qH9V7XabHVJ0jzpLSyq6hngqSSntNI64BFgBzA9o2kjcHt
b3gFc3GZFnQW80Iar7gTWJ1nebmyvbzVJ0jw5pufj/wrwe0mOBZ4ALmEQULck2QQ8CVzY2t4BnAdMAi+1t
lTVviRXAPe2dpdX1b6e+y1JGtJrWFTVA8D4DJvWzdC2gEtnOc42YNucdk6SNDJ/wS1J6mRYSJI6GRaSpE6Gh
SSpk2EhSepkWEiSOhkWkqROhoUkqZNhIUnqZFhIkjoZFpKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhS
erUa1gk+XqSryZ5IMlEqx2fZGeSx9v38lZPkmuSTCZ5MMnpQ8fZ2No/nmTjbOeTJPVjPq4s/npVnVZV069Xv
QzYVVVrgV1tHeBcYG37bAauhUG4AFuAM4EzgC3TASNJmh8LMQy1AdjelrcD5w/Vb6iBu4BlSU4CzgZ2VtW
+qtoP7ATOmec+S9KS1ndYFPDfk9yXZHOrraiqp9vyM8CKtrwSeGpo392tNlv9VZJsTjKRZGJqamou/wZJWvK
O6fn4f62q9iT5YWBnkj8Z3lhVlaTm4kRVtRXYCjA+Pj4nx5QkDfR6ZVFVe9r3XuA2Bvccnm3DS7Tvva35HmD1
0O6rWm22uiRpnvQWFknemuTt08vAeuAhYAcwPaNpI3B7W94BXNxmRZ0FvNCGq+4E1idZ3m5sr281SdI8
6XMYagVwW5Lp8/ynqvpcknuBW5JsAr4GXNXmmRZ0FvNCGq+4E1idZ3m5sr281SdI8
6XMYagVwW5Lp8/ynqvpcknuBW5JsAp4ELmzt7wDOAyaBl4BLAKpqX5IrgHtbu8ural+P/ZYkHaS3sKiqJ4B3z
1B/Dlg3Q72AS2c51jZg21z3UZI0Gn/BLUnqZFhIkjoZFpKkToaFJKnTSGGRZNcoNUnS69MhZ0MleRPwFuDE9j
huHtE0/xAyP3JAkvT51TZ39ReDDwI8A9/G9sPgm8O/665YkaTE5ZFhU1dXA1Ul+pao+OU99kiSNMiP9KK+qPpnkrwBrhvepqht66pckaREZKSyS3Aj8ZeAB4OVWLsCwkKQlYNTHfYwDp7YHAEqSlphRf2fxEPAX++yIJGnxGvXK4kTgkST3AAemi1X1s730SpK0qIwaFv+8z05Ikha3UWdD/WHfHZEkLV6jzob6FoPZTwDHAscA366qd/TVMUnS4jHqlcXbp5czeG/qBuCsvjolSVpcDvups+1Nd/8VOLuH/kiSFqFRh6F+bmj1DQx+d/H/eumRJGnRGXU21N8aWn4J+CqDoShJ0hIw6j2LS/ruiCRp8Rr15UerktyWZG/7fDrJqr47J0laHEa9wf27wA4G77X4EeC/tZokaQkYNSzGqup3q+ql9rkeGOuxX5KkRWTUsHguyc8nOap9fh54rs+OSZIWj1HD4heBC4FngKeB9wK/MMqOLVzuT/KZtn5ykruTTCb5/STHtvob2/pk275m6Bgfb/XHkjgLS5Lm2ahhcTmwsarGquqHGYTHvxhx3w8Bjw6t/yZwVVX9KLAf2NTqm4D9rX5Va0eSU4GLgHcC5wC/k+SoEc8tSZoDo4bFu6pq//RKVe0D3vNaO7UZU38T+FRbD/AzwK2tyXbg/La8oa3Ttq8beqzIzVV1oKq+AkwCZ4zYb0nSHBg1LN6QZPn0SpLjGe03Gv8O+CfAd9v6CcDzVfVSW98NrGzLK4GnANr2F1r7V+oz7CNJmgej/oL73wBfTPKf2/oFwJWH2iHJ3wb2VtV9SX76+9PQAQAAAIlJREFUe+/iaJJsBjYDvOMd7+j7dJK0pIz6C+4bkkwwGEIC+LmqeuQ1dvurwM8mOQ94E/AO4GpgWZKj29XDKmBPa78HWA3sTnI0cByDGVfT9WnD+wz3cSuwFWB8fLwO3i5J+t6N/NTZqnqkqv59+7xWUFBVH6+qVVW1hsEN6s9X1fuBLzCYTQWwEbi9Le9o67Ttn6+qavWL2mypk4G1wD2j9luS9P0bdRhqLn0MuDnJbwD3A9e1+nXAjUkmgX0MAoaqejjJLcAjDB5ieGlVvTz/3ZakpWtewqKq/gD4g7b8BDPMZqqq/8fgXshM+1/Ja9wjkST157BffiRJWnoMC0lSJ8NCktTJsJAkdTIsJEmdDAtJUifDQpLUybCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0lSJ8NCktTJsJAkdTIsJEmdDAtJUifDQpLUybCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ16C4skb0pyT5IvJ3k4yb9o9ZOT3J1kMsnvJzm21d/Y1ifb9jVDx/p4qz+W5Oy++ixJmlmfVxYHgJ+pqncDpwHnJDkL+E3gqqr6UWA/sKm13wTsb/WrWjuSnApcBLwTOAf4nSRH9dhvSdJBeguLGnixrR7TPgX8DHBrq28Hzm/LG9o6bfu6JGn1m6vqQFV9BZgEzuir35KkV+v1nkWSo5I8AOwFdgJ/CjxfVS+1JruBlW15JfAUQNv+AnDCcH2GfSRJ86DXsKiql6vqNGAVg6uBH+vrXEk2J5lIMjE1NdXXaSRpSZqX2VBV9TzwBeCngGVJjm6bVgF72vIeYDVA234c8NxwfYZ9hs+xtarGq2p8bGysl79DkpaqPmdDjSVZ1pbfDPwN4FEGofHe1mwjcHtb3tHWads/X1XV6he12VInA2uBe/rqtyTp1Y5+7Sbfs5OA7W3m0huAW6rqM0keAW5O8hvA/cB1rf11wI1JJoF9DGZAUVUPJ7kFeAR4Cbi0ql7usd+SpIP0FhZV9SDwnhnqTzDDbKaq+n/ABbMc60rgyrnuoyRpNP6CW5LUybCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0lSJ8NCktTJsJAkdTIsJEmdDAtJUifDQpLUybCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0lSJ8NCktTJsJAkdTIsJEmdDAtJUqfewiLJ6iRfSPJIkoeTfKjVj0+yM8nj7Xt5qyfJNUkmkzyY5PShY21s7R9PsrGvPkuSZtbnlcVLwK9W1anAWcClSU4FLgN2VdVaYFdbBzgXWNs+m4FrYRAuwBbgTOAMYMt0wEiS5kdvYVFVT1fVl9ryt4BHgZXABmB7a7YdOL8tbwBuqIG7gGVJTgLOBnZW1b6q2g/sBM7pq9+SpFebl3sWSdYA7wHuBlZU1dNt0zPAira8EnhqaLfdrTZb/eBzbE4ykWRiampqTvsvSUtd72GR5G3Ap4EPV9U3h7dVVQE1F+epqq1VNV5V42NjY3NxSElS02tYJDmGQVD8XlX9l1Z+tg0v0b73tvoeYPXQ7qtabba6JGme9DkbKsB1wKNV9W+HNu0Apmc0bQRuH6pf3GZFnQW80Iar7gTWJ1nebmyvbzVJ0jw5usdj/1Xg7wFfTfJAq/0a8AngliSbgK8BF7ZtdwDnAZPAS8AlAFW1L8kVwL2t3eVVta/HfkuSDtJbWFTV/wAyy+Z1M7Qv4NJZjrUN2DZ3vZMkHQ5/wS1J6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhIUnqZFhIkjoZFpKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhIUnqZFhIkjoZFpKkToaFJKlTb2GRZFuSvUkeGqodn2Rnksfb9/JWT5JrkkwmeTDJ6UP7bGztH0+ysa/+SpJm1+eVxfXAOQfVLgN2VdVaYFdbBzgXWNs+m4FrYRAuwBbgTOAMYMt0wEiS5k9vYVFVfwTsO6i8AdjelrcD5w/Vb6iBu4BlSU4CzgZ2VtW+qtoP7OTVASRJ6tl837NYUVVPt+VngBVteSXw1FC73a02W/1VkmxOMpFkYmpqam57LUlL3ILd4K6qAmoOj7e1qsaranxsbGyuDitJYv7D4tk2vET73tvqe4DVQ+1Wtdps9UlSfNkPsLig3xvCApgBzA9o2kjcHt7vhr1bftucuGQpKlg8tdq9oq+DxafBLa0j4/VVV/WJi8fyerqxmu8Xr9H4pBXUTB1b+XfvlWn1ElvvDtNj9VGVt1S9eZ/nQKlt9H3k+3oU4bMYDo/p75n7OfKeJtdiuMdf0m3Vxm0PJPt4xXXBbVlZaJ2/xrl0vzBH2gUqz8j7lBW8fxLjBfXH3X7jbTWwmlF1/yXE9YjJcLQ/f2KX5RW1x3Pq3jzX3Nrn1xZZtlp/c/n1f5cfG0rp+Jcu/78x9vOHqrtaPNjzf6fmVLRPkdjLf25jB/6cVdLN5dpHwlg/q2MreyzNLetcjVe4Zmcc+4dVm+oXXnvnm8t+Z4ef6QhnN6t7dd9zXD86cN/0vyTRXV1TOexX8e7ve8fS/LM8ZZ/QHrvr64X4D6XJWkWVn8vmtt5s7X9Y92NVXNqcWe6j8v/y2ZyM6Tzr2ce11Tq3uWnvQ8ufXQT84f7eOFmZ/nHx/xFsmvu7WV8W8X/2yWXcFzemrZtS85j63Nw9bqmZhhd5vXxT/5qsu7/6+X0OnB+iFgAAAIpJREFU6fny+u/sLT/qTr+fc7Y/dZ7Wq7jX84/i7g5cjs+tTzcjnTzdL5q+oKWt6Yg+tCmHHx/p3BMvcg/2eabplrHz5lfN6F+r9m7o+tfObPKbb4VD1vYn+o++3+ln79vPz3Xf6fZn9L73m05bfNf9f5o+75+pv1a5Z6ff+6rvWbbvuJ86vh8uubrs6T/lqr89rj+v/J3v+Xf9+YN6+Pd+Efwr//jG/+OpmHn6L/+Uf+hPfOHTpp67r6+dvfr+/7+6X/Jw3P4u+qOtfX+6vpz9D56/Ck7c8Fvw69Q6+LsXZ/f9+zjc2vO+Zd8J/u+B1zl63mdH/lnz8N2n+9N8dHE8f/ZR/M+89tTuqD9yd8A

```
SVInw0KS1KnPd3CvTvKFJI8keTjJr7b68Ul2Jnm8fS9v9SS5JslkkgeTnD50rI2t/eNJNs52TklSP/q8sjgA/OOqOh
U4C7g0yanAZcCuqloL7GrrAOcCa9tnM3AtDMIF2AKcyeB1rFumA0aSND/6fAf308DTbfnFJI8CK4ENwM+0Zt
uBLwK/0eo3VFUBdyVZluSk1nZnVe0DSLITOAe4sa++S4vZNy7/ywvdBS1C7/iXX+31+PNyzyLJGuA9wN3AihY
kAM8AK9rySuCpod12t9ps9YPPsTnJRJKJqampuf0DJGmJ6z0skrwN+H3gw1X1zeFt7Sqi5uI8VbW1qsaranxs
bGwuDilJanoNiyRvYBAUv1dVf9DKz7bhJdr33lbfA6we2n1Vq81WlyTNkz5nQwW4Dni0qv7N0KYdwPSMpo
3A7UP1i9usqLOAF9pw1Z3A+iTL243t9a0mSZonvd3gBt4L/D3gq0keaLV/BnwMuCXJJuBJ4MK27Q7gPGASe
Am4BKCq9iW5Ari3tbt8+ma3JGl+9Dkb6o+AzLJ53QztC7h0lmNtA7bNXe8kSd8Pf8EtSepkWEiSOhkWkqROh
oUkqZNhIUnqZFhIkjoZFpKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhI
Unq1Oc7uLcl2ZvkoaHa8Ul2Jnm8fS9v9SS5JslkkgeTnD60z8bW/vEkG2c6lySpX31eWXwaOOeg2mXArqpaC
+xq6wDnAmvbZzNwLQzCBdgCnAmcAWyZDhhJ0vzpLSyq6kvAvoPKG4DtbXk7cP5Q/YYauAtYluQk4GxgZ1X
tq6r9wE5eG0CSpJ7N9z2LFVX1dFt+BljRllcCTw21291qs9VfI8nmJBNJJqampua215K0xC3YDe6qKqDm8Hhb
q2q8qsbHxsbm6rCSJOY/LJ5tw0u0772tvgdYPdRuVavNVpckzaP5DosdwPSMpo3A7UP1i9usqLOAF9pw1Z3
A+iTL243t9a0mSZpHx/R14CQ3Aj8DnJhkN4NZTR8DbkmyCXgSuLA1vwM4D5gEXgIuAaiqfUmuAO5t7S6vqo
NvmkuSetZbWFTVB2fZtG6GtgVcOstxtgHb5rBrkqTvk7/gliR1MiwkSZ0MC0lSJ8NCktTJsJAkdTIsJEmdDAtJUif
DQpLUybCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0lSJ8NCktTJsJAkdTpiwiLJOUkeSzKZ5LKF7o
8kLSVHRFgkORr498C5wKnAB5OcurC9kqSl44gIC+AMYLKqnqiqPwduAjYscJ8kack4ZqE7MKKVwFND67uBM
4cbJNkMbG6r30ry2Dz1bSk4Efizhe7EYpBPbFzoLujV/Lc5bUvm4ih/cbYNR0pYdKqqrcDWhe7H61GSiaoaX+h
+SAfz3+b8OVKGofYAq4fWV7WaJGkeHClhcS+wNsnJSY4FLgJ2LHCfJGnJOCKGoarqQJJfBu4Ejga2VdXDC9ytp
cThPS1W/tucJ6mqhe6DJGmRO1KGoSRJC8iwkCR1Mix0SD5mRYtRkm1J9iZ5aKH7slQYFpqVj1nRIvZp4JyF7s
RSYljoUHzMihalqvoSsG+h+7GUGBY6lJkes7JygfoiaQEZFpKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhI
5VHgFh+zosUgyY3Al4FTkuxOsmmh+/R65+M+JEmdvLKQJHUyLCRJnQwLSVInw0KS1MmwkCR1Miykw5Dkl
SQPDH0O+UTeJHckWdWdY+v3QY5/vNJP/k8Hss/WCOiNeqSovQ/62q00ZtXFXnASRZA/wS8Kl+uiX1wysLaY4k
Oa69++OUtn5jkn/Ylr+e5ETgY8CPtquR32nbPpLk3iQPJvmtoeP98yT/J8kfAacswJ8kfZdXFtLheXOSB4bW/1V
V3Zzkl4FPJ7kaWF5V//Gg/S4DfmL4Gvv7+HOnQDAvp8Mw4DF
VVO5NcwOClUe8e4Tjr2+f+tv42BuHxduC2qnoJIInP5NKCchhKmkNJjgJ+HHgJWD7KLgyuSk5rn3dW1XW9dlI
6DIaFNLd+jcFDF/8OcH2SNxy0/UUGVw3T7gR+IcnbAJKsTPLDwJeA85O8OcnbgZ/rv+vS7ByGkg7PwfcsPgdc
D/wD4IyqejHJl4B/AWyZblRVzyX53+2Gtv+VzAv5bVX0kyY8DX04C8C3gQ1X1lSQ3A38M7GXwuHhpwfjUWUlSJ4e
hJEmdDAtJUifDQpLUybCQJHUyLCRJnQwLSVInw0KS1On/A+2Z0j36VccUAAAAAElFTkSuQmCC\n",
    "text/plain": [
     "<Figure size 432x288 with 1 Axes>"
    ]
   },
   "metadata": {
    "needs_background": "light"
   },
   "output_type": "display_data"
  }
 ],
  "source": [
   "sns.countplot(df['Exited'])"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 18,
  "id": "c77c724c",
  "metadata": {
   "execution": {
    "iopub.execute_input": "2021-12-21T16:51:28.715665Z",
    "iopub.status.busy": "2021-12-21T16:51:28.714984Z",
    "iopub.status.idle": "2021-12-21T16:51:28.718504Z",
    "shell.execute_reply": "2021-12-21T16:51:28.717824Z",
    "shell.execute_reply.started": "2021-12-21T16:22:23.241922Z"
   },
   "papermill": {
    "duration": 0.044497,
    "end_time": "2021-12-21T16:51:28.718649",
    "exception": false,
    "start_time": "2021-12-21T16:51:28.674152",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
     "0    7963\n",
     "1    2037\n",
     "Name: Exited, dtype: int64"
    ]
   },
   "execution_count": 18,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "df['Exited'].value_counts()"
 ]
},
{
 "cell_type": "code",
 "execution_count": 19,
 "id": "f4ba6a1f",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:51:28.793562Z",
   "iopub.status.busy": "2021-12-21T16:51:28.792872Z",
   "iopub.status.idle": "2021-12-21T16:51:28.798414Z",
   "shell.execute_reply": "2021-12-21T16:51:28.797875Z",
   "shell.execute_reply.started": "2021-12-21T16:22:23.251339Z"
  },
  "papermill": {
   "duration": 0.045252,
   "end_time": "2021-12-21T16:51:28.798555",
     "exception": false,
     "start_time": "2021-12-21T16:51:28.753303",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "#splitting data to input and output\n",
    "X=df.drop('Exited',axis=1) #input\n",
    "y=df['Exited']  #output(label)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 20,
   "id": "2257d50b",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:28.875644Z",
     "iopub.status.busy": "2021-12-21T16:51:28.874865Z",
     "iopub.status.idle": "2021-12-21T16:51:28.880854Z",
     "shell.execute_reply": "2021-12-21T16:51:28.880316Z",
     "shell.execute_reply.started": "2021-12-21T16:22:23.362068Z"
    },
    "papermill": {
     "duration": 0.047357,
     "end_time": "2021-12-21T16:51:28.881009",
     "exception": false,
     "start_time": "2021-12-21T16:51:28.833652",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "X_train, X_test, y_train, y_test= train_test_split(X,y,test_size=0.2,shuffle=True)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 21,
   "id": "0799924a",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:28.955883Z",
     "iopub.status.busy": "2021-12-21T16:51:28.955093Z",
     "iopub.status.idle": "2021-12-21T16:51:28.958213Z",
     "shell.execute_reply": "2021-12-21T16:51:28.958731Z",
     "shell.execute_reply.started": "2021-12-21T16:22:23.589379Z"
    },
    "papermill": {
     "duration": 0.04341,
     "end_time": "2021-12-21T16:51:28.958900",
     "exception": false,
     "start_time": "2021-12-21T16:51:28.915490",
     "status": "completed"
    },
   "tags": []
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "(10000, 13)\n",
     "(10000,)\n"
    ]
   }
  ],
  "source": [
   "print(X.shape)\n",
   "print(y.shape)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 22,
  "id": "a7e5734c",
  "metadata": {
   "execution": {
    "iopub.execute_input": "2021-12-21T16:51:29.034485Z",
    "iopub.status.busy": "2021-12-21T16:51:29.033527Z",
    "iopub.status.idle": "2021-12-21T16:51:29.038550Z",
    "shell.execute_reply": "2021-12-21T16:51:29.037927Z",
    "shell.execute_reply.started": "2021-12-21T16:22:23.955086Z"
   },
    "papermill": {
     "duration": 0.045016,
     "end_time": "2021-12-21T16:51:29.038692",
     "exception": false,
     "start_time": "2021-12-21T16:51:28.993676",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      " X_train.shape :  (8000, 13)\n",
      " y_train.shape :  (8000,)\n",
      " X_test.shape :  (2000, 13)\n",
      " y_test.shape :  (2000,)\n"
     ]
    }
   ],
   "source": [
    "print(' X_train.shape : ',X_train.shape)\n",
    "print(' y_train.shape : ',y_train.shape)\n",
    "print(' X_test.shape : ',X_test.shape)\n",
    "print(' y_test.shape : ',y_test.shape)"
   ]
  },
  {
   "cell_type": "markdown",
   "id": "d53cfe60",
   "metadata": {
    "papermill": {
     "duration": 0.034608,
     "end_time": "2021-12-21T16:51:29.108819",
     "exception": false,
     "start_time": "2021-12-21T16:51:29.074211",
     "status": "completed"
    },
    "tags": []
   },
   "source": [
    "# deep learning ANN"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 23,
   "id": "f2ddcb87",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:29.186141Z",
     "iopub.status.busy": "2021-12-21T16:51:29.185171Z",
     "iopub.status.idle": "2021-12-21T16:51:29.317013Z",
     "shell.execute_reply": "2021-12-21T16:51:29.317542Z",
     "shell.execute_reply.started": "2021-12-21T16:22:24.415566Z"
    },
    "papermill": {
```

```
   "duration": 0.174048,
   "end_time": "2021-12-21T16:51:29.317717",
   "exception": false,
   "start_time": "2021-12-21T16:51:29.143669",
   "status": "completed"
  },
  "tags": []
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "Model: \"sequential\"\n",
    "_________________________________________________________________\n",
    "Layer (type)                 Output Shape              Param #   \n",
    "=================================================================\n",
    "dense (Dense)                (None, 6)                 84        \n",
    "_________________________________________________________________\n",
    "dense_1 (Dense)              (None, 5)                 35        \n",
    "_________________________________________________________________\n",
    "dense_2 (Dense)              (None, 1)                 6         \n",
    "=================================================================\n",
    "Total params: 125\n",
    "Trainable params: 125\n",
    "Non-trainable params: 0\n",
    "_________________________________________________________________\n"
   ]
  },
{
 "name": "stderr",
 "output_type": "stream",
 "text": [
  "\n",
  "User settings:\n",
  "\n",
  "   KMP_AFFINITY=granularity=fine,verbose,compact,1,0\n",
  "   KMP_BLOCKTIME=0\n",
  "   KMP_DUPLICATE_LIB_OK=True\n",
  "   KMP_INIT_AT_FORK=FALSE\n",
  "   KMP_SETTINGS=1\n",
  "   KMP_WARNINGS=0\n",
  "\n",
  "Effective settings:\n",
  "\n",
  "   KMP_ABORT_DELAY=0\n",
  "   KMP_ADAPTIVE_LOCK_PROPS='1,1024'\n",
  "   KMP_ALIGN_ALLOC=64\n",
  "   KMP_ALL_THREADPRIVATE=128\n",
  "   KMP_ATOMIC_MODE=2\n",
  "   KMP_BLOCKTIME=0\n",
  "   KMP_CPUINFO_FILE: value is not defined\n",
  "   KMP_DETERMINISTIC_REDUCTION=false\n",
  "   KMP_DEVICE_THREAD_LIMIT=2147483647\n",
  "   KMP_DISP_NUM_BUFFERS=7\n",
  "   KMP_DUPLICATE_LIB_OK=true\n",
  "   KMP_ENABLE_TASK_THROTTLING=true\n",
  "   KMP_FORCE_REDUCTION: value is not defined\n",
"   KMP_FOREIGN_THREADS_THREADPRIVATE=true\n",
"   KMP_FORKJOIN_BARRIER='2,2'\n",
"   KMP_FORKJOIN_BARRIER_PATTERN='hyper,hyper'\n",
"   KMP_GTID_MODE=3\n",
"   KMP_HANDLE_SIGNALS=false\n",
"   KMP_HOT_TEAMS_MAX_LEVEL=1\n",
"   KMP_HOT_TEAMS_MODE=0\n",
"   KMP_INIT_AT_FORK=true\n",
"   KMP_LIBRARY=throughput\n",
"   KMP_LOCK_KIND=queuing\n",
"   KMP_MALLOC_POOL_INCR=1M\n",
"   KMP_NUM_LOCKS_IN_BLOCK=1\n",
"   KMP_PLAIN_BARRIER='2,2'\n",
"   KMP_PLAIN_BARRIER_PATTERN='hyper,hyper'\n",
"   KMP_REDUCTION_BARRIER='1,1'\n",
"   KMP_REDUCTION_BARRIER_PATTERN='hyper,hyper'\n",
"   KMP_SCHEDULE='static,balanced;guided,iterative'\n",
"   KMP_SETTINGS=true\n",
"   KMP_SPIN_BACKOFF_PARAMS='4096,100'\n",
"   KMP_STACKOFFSET=64\n",
"   KMP_STACKPAD=0\n",
"   KMP_STACKSIZE=8M\n",
"   KMP_STORAGE_MAP=false\n",
"   KMP_TASKING=2\n",
"   KMP_TASKLOOP_MIN_TASKS=0\n",
"   KMP_TASK_STEALING_CONSTRAINT=1\n",
"   KMP_TEAMS_THREAD_LIMIT=4\n",
"   KMP_TOPOLOGY_METHOD=all\n",
"   KMP_USE_YIELD=1\n",
    "   KMP_VERSION=false\n",
    "   KMP_WARNINGS=false\n",
    "   OMP_AFFINITY_FORMAT='OMP: pid %P tid %i thread %n bound to OS proc set {%A}'\n",
    "   OMP_ALLOCATOR=omp_default_mem_alloc\n",
    "   OMP_CANCELLATION=false\n",
    "   OMP_DEFAULT_DEVICE=0\n",
    "   OMP_DISPLAY_AFFINITY=false\n",
    "   OMP_DISPLAY_ENV=false\n",
    "   OMP_DYNAMIC=false\n",
    "   OMP_MAX_ACTIVE_LEVELS=1\n",
    "   OMP_MAX_TASK_PRIORITY=0\n",
    "   OMP_NESTED: deprecated; max-active-levels-var=1\n",
    "   OMP_NUM_THREADS: value is not defined\n",
    "   OMP_PLACES: value is not defined\n",
    "   OMP_PROC_BIND='intel'\n",
    "   OMP_SCHEDULE='static'\n",
    "   OMP_STACKSIZE=8M\n",
    "   OMP_TARGET_OFFLOAD=DEFAULT\n",
    "   OMP_THREAD_LIMIT=2147483647\n",
    "   OMP_WAIT_POLICY=PASSIVE\n",
    "   KMP_AFFINITY='verbose,warnings,respect,granularity=fine,compact,1,0'\n",
    "\n",
    "2021-12-21 16:51:29.218493: I tensorflow/core/common_runtime/process_util.cc:146] Creating new thread pool with default inter op setting: 2. Tune using inter_op_parallelism_threads for best performance.\n"
   ]
  }
 ],
 "source": [
    "model = Sequential()\n",
    "model.add(Dense(6, input_dim=13, activation='relu'))\n",
    "\n",
    "model.add(Dense(5, activation='relu'))\n",
    "\n",
    "model.add(Dense(1, activation='sigmoid'))\n",
    "\n",
    "model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])\n",
    "model.summary()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 24,
   "id": "7bfd3f2e",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:51:29.396105Z",
     "iopub.status.busy": "2021-12-21T16:51:29.395447Z",
     "iopub.status.idle": "2021-12-21T16:53:52.009314Z",
     "shell.execute_reply": "2021-12-21T16:53:52.008618Z",
     "shell.execute_reply.started": "2021-12-21T16:22:24.612490Z"
    },
    "papermill": {
     "duration": 142.65612,
     "end_time": "2021-12-21T16:53:52.009498",
     "exception": false,
     "start_time": "2021-12-21T16:51:29.353378",
     "status": "completed"
   },
   "tags": []
  },
  "outputs": [
   {
    "name": "stderr",
    "output_type": "stream",
    "text": [
     "2021-12-21 16:51:29.482130: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)\n"
    ]
   },
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Epoch 1/100\n",
     "680/680 [==============================] - 2s 2ms/step - loss: 0.4925 - accuracy: 0.7728 - val_loss: 0.4564 - val_accuracy: 0.7825\n",
     "Epoch 2/100\n",
     "680/680 [==============================] - 1s 2ms/step - loss: 0.4351 - accuracy: 0.8031 - val_loss: 0.4400 - val_accuracy: 0.8033\n",
     "Epoch 3/100\n",
     "680/680 [==============================] - 1s 2ms/step - loss: 0.4233 - accuracy: 0.8143 - val_loss: 0.4335 - val_accuracy: 0.8150\n",
     "Epoch 4/100\n",
     "680/680 [==============================] - 1s 2ms/step - loss: 0.4164 - accuracy: 0.8216 - val_loss: 0.4302 - val_accuracy: 0.8233\n",
     "Epoch 5/100\n",
     "680/680 [==============================] - 1s 2ms/step - loss: 0.4121 - accuracy: 0.8254 - val_loss: 0.4280 - val_accuracy: 0.8275\n",
    "Epoch 6/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.4086 - accuracy: 0.8297 - val_loss: 0.4246 - val_accuracy: 0.8283\n",
    "Epoch 7/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.4057 - accuracy: 0.8331 - val_loss: 0.4229 - val_accuracy: 0.8292\n",
    "Epoch 8/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.4028 - accuracy: 0.8343 - val_loss: 0.4192 - val_accuracy: 0.8275\n",
    "Epoch 9/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3997 - accuracy: 0.8338 - val_loss: 0.4177 - val_accuracy: 0.8317\n",
    "Epoch 10/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3960 - accuracy: 0.8338 - val_loss: 0.4149 - val_accuracy: 0.8333\n",
    "Epoch 11/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3930 - accuracy: 0.8372 - val_loss: 0.4096 - val_accuracy: 0.8383\n",
    "Epoch 12/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3877 - accuracy: 0.8379 - val_loss: 0.4071 - val_accuracy: 0.8367\n",
    "Epoch 13/100\n",
    "680/680 [==============================] - 2s 2ms/step - loss: 0.3827 - accuracy: 0.8397 - val_loss: 0.3988 - val_accuracy: 0.8383\n",
    "Epoch 14/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3781 - accuracy: 0.8413 - val_loss: 0.3974 - val_accuracy: 0.8342\n",
    "Epoch 15/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3745 - accuracy: 0.8441 - val_loss: 0.3948 - val_accuracy: 0.8367\n",
    "Epoch 16/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3719 - accuracy: 0.8447 - val_loss: 0.3936 - val_accuracy: 0.8358\n",
    "Epoch 17/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3703 - accuracy: 0.8421 -
val_loss: 0.3941 - val_accuracy: 0.8383\n",
    "Epoch 18/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3680 - accuracy: 0.8431 -
val_loss: 0.3911 - val_accuracy: 0.8375\n",
    "Epoch 19/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3664 - accuracy: 0.8453 -
val_loss: 0.3910 - val_accuracy: 0.8342\n",
    "Epoch 20/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3653 - accuracy: 0.8437 -
val_loss: 0.3918 - val_accuracy: 0.8325\n",
    "Epoch 21/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3641 - accuracy: 0.8443 -
val_loss: 0.3927 - val_accuracy: 0.8383\n",
    "Epoch 22/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3632 - accuracy: 0.8449 -
val_loss: 0.3913 - val_accuracy: 0.8400\n",
    "Epoch 23/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3616 - accuracy: 0.8456 -
val_loss: 0.3917 - val_accuracy: 0.8400\n",
    "Epoch 24/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3615 - accuracy: 0.8440 -
val_loss: 0.3899 - val_accuracy: 0.8375\n",
    "Epoch 25/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3602 - accuracy: 0.8462 -
val_loss: 0.3915 - val_accuracy: 0.8367\n",
    "Epoch 26/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3599 - accuracy: 0.8468 -
val_loss: 0.3892 - val_accuracy: 0.8342\n",
    "Epoch 27/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3590 - accuracy: 0.8456 -
val_loss: 0.3913 - val_accuracy: 0.8400\n",
    "Epoch 28/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3585 - accuracy: 0.8484 -
val_loss: 0.3876 - val_accuracy: 0.8342\n",
    "Epoch 29/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3585 - accuracy: 0.8460 -
val_loss: 0.3872 - val_accuracy: 0.8350\n",
    "Epoch 30/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3576 - accuracy: 0.8456 -
val_loss: 0.3881 - val_accuracy: 0.8383\n",
    "Epoch 31/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3578 - accuracy: 0.8463 -
val_loss: 0.3885 - val_accuracy: 0.8383\n",
    "Epoch 32/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3577 - accuracy: 0.8479 -
val_loss: 0.3931 - val_accuracy: 0.8342\n",
    "Epoch 33/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3570 - accuracy: 0.8476 -
val_loss: 0.3902 - val_accuracy: 0.8367\n",
    "Epoch 34/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3569 - accuracy: 0.8463 -
val_loss: 0.3898 - val_accuracy: 0.8342\n",
    "Epoch 35/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3561 - accuracy: 0.8485 -
val_loss: 0.3882 - val_accuracy: 0.8375\n",
    "Epoch 36/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3563 - accuracy: 0.8479 -
val_loss: 0.3863 - val_accuracy: 0.8375\n",
    "Epoch 37/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3562 - accuracy: 0.8478 -
val_loss: 0.3875 - val_accuracy: 0.8417\n",
    "Epoch 38/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3561 - accuracy: 0.8469 -
val_loss: 0.3873 - val_accuracy: 0.8375\n",
    "Epoch 39/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3552 - accuracy: 0.8500 -
val_loss: 0.3916 - val_accuracy: 0.8342\n",
    "Epoch 40/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3555 - accuracy: 0.8493 -
val_loss: 0.3871 - val_accuracy: 0.8367\n",
    "Epoch 41/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3544 - accuracy: 0.8482 -
val_loss: 0.3879 - val_accuracy: 0.8350\n",
    "Epoch 42/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3549 - accuracy: 0.8491 -
val_loss: 0.3858 - val_accuracy: 0.8367\n",
    "Epoch 43/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3545 - accuracy: 0.8491 -
val_loss: 0.3898 - val_accuracy: 0.8342\n",
    "Epoch 44/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3542 - accuracy: 0.8471 -
val_loss: 0.3873 - val_accuracy: 0.8358\n",
    "Epoch 45/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3537 - accuracy: 0.8497 -
val_loss: 0.3891 - val_accuracy: 0.8383\n",
    "Epoch 46/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3532 - accuracy: 0.8490 -
val_loss: 0.3881 - val_accuracy: 0.8375\n",
    "Epoch 47/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3530 - accuracy: 0.8479 -
val_loss: 0.3869 - val_accuracy: 0.8350\n",
    "Epoch 48/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3522 - accuracy: 0.8506 -
val_loss: 0.3870 - val_accuracy: 0.8350\n",
    "Epoch 49/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3523 - accuracy: 0.8506 -
val_loss: 0.3849 - val_accuracy: 0.8375\n",
    "Epoch 50/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3517 - accuracy: 0.8507 - val_loss: 0.3862 - val_accuracy: 0.8358\n",
    "Epoch 51/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3513 - accuracy: 0.8525 - val_loss: 0.3857 - val_accuracy: 0.8375\n",
    "Epoch 52/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3509 - accuracy: 0.8519 - val_loss: 0.3884 - val_accuracy: 0.8308\n",
    "Epoch 53/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3509 - accuracy: 0.8531 - val_loss: 0.3849 - val_accuracy: 0.8367\n",
    "Epoch 54/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3501 - accuracy: 0.8538 - val_loss: 0.3838 - val_accuracy: 0.8367\n",
    "Epoch 55/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3494 - accuracy: 0.8534 - val_loss: 0.3846 - val_accuracy: 0.8392\n",
    "Epoch 56/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3487 - accuracy: 0.8528 - val_loss: 0.3804 - val_accuracy: 0.8392\n",
    "Epoch 57/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3482 - accuracy: 0.8522 - val_loss: 0.3789 - val_accuracy: 0.8392\n",
    "Epoch 58/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3475 - accuracy: 0.8556 - val_loss: 0.3781 - val_accuracy: 0.8433\n",
    "Epoch 59/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3484 - accuracy: 0.8531 - val_loss: 0.3784 - val_accuracy: 0.8433\n",
    "Epoch 60/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3467 - accuracy: 0.8557 - val_loss: 0.3827 - val_accuracy: 0.8417\n",
    "Epoch 61/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3472 - accuracy: 0.8574 - val_loss: 0.3764 - val_accuracy: 0.8433\n",
    "Epoch 62/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3454 - accuracy: 0.8546 - val_loss: 0.3772 - val_accuracy: 0.8417\n",
    "Epoch 63/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3455 - accuracy: 0.8549 - val_loss: 0.3791 - val_accuracy: 0.8433\n",
    "Epoch 64/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3442 - accuracy: 0.8554 - val_loss: 0.3735 - val_accuracy: 0.8442\n",
    "Epoch 65/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3435 - accuracy: 0.8568 - val_loss: 0.3709 - val_accuracy: 0.8483\n",
    "Epoch 66/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3437 - accuracy: 0.8578 - val_loss: 0.3715 - val_accuracy: 0.8508\n",
    "Epoch 67/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3427 - accuracy: 0.8547 - val_loss: 0.3702 - val_accuracy: 0.8483\n",
    "Epoch 68/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3423 - accuracy: 0.8571 - val_loss: 0.3702 - val_accuracy: 0.8483\n",
    "Epoch 69/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3418 - accuracy: 0.8556 - val_loss: 0.3688 - val_accuracy: 0.8467\n",
    "Epoch 70/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3409 - accuracy: 0.8576 - val_loss: 0.3700 - val_accuracy: 0.8458\n",
    "Epoch 71/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3416 - accuracy: 0.8571 - val_loss: 0.3732 - val_accuracy: 0.8517\n",
    "Epoch 72/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3417 - accuracy: 0.8574 -
val_loss: 0.3686 - val_accuracy: 0.8500\n",
    "Epoch 73/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3406 - accuracy: 0.8581 -
val_loss: 0.3681 - val_accuracy: 0.8525\n",
    "Epoch 74/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3401 - accuracy: 0.8576 -
val_loss: 0.3646 - val_accuracy: 0.8500\n",
    "Epoch 75/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3391 - accuracy: 0.8565 -
val_loss: 0.3672 - val_accuracy: 0.8533\n",
    "Epoch 76/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3403 - accuracy: 0.8579 -
val_loss: 0.3672 - val_accuracy: 0.8483\n",
    "Epoch 77/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3397 - accuracy: 0.8563 -
val_loss: 0.3698 - val_accuracy: 0.8533\n",
    "Epoch 78/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3388 - accuracy: 0.8600 -
val_loss: 0.3712 - val_accuracy: 0.8525\n",
    "Epoch 79/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3383 - accuracy: 0.8579 -
val_loss: 0.3648 - val_accuracy: 0.8558\n",
    "Epoch 80/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3367 - accuracy: 0.8588 -
val_loss: 0.3693 - val_accuracy: 0.8542\n",
    "Epoch 81/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3376 - accuracy: 0.8600 -
val_loss: 0.3637 - val_accuracy: 0.8575\n",
    "Epoch 82/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3374 - accuracy: 0.8601 -
val_loss: 0.3639 - val_accuracy: 0.8558\n",
    "Epoch 83/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3365 - accuracy: 0.8610 - val_loss: 0.3641 - val_accuracy: 0.8558\n",
    "Epoch 84/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3359 - accuracy: 0.8618 - val_loss: 0.3616 - val_accuracy: 0.8558\n",
    "Epoch 85/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3363 - accuracy: 0.8609 - val_loss: 0.3612 - val_accuracy: 0.8517\n",
    "Epoch 86/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3353 - accuracy: 0.8600 - val_loss: 0.3604 - val_accuracy: 0.8517\n",
    "Epoch 87/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3354 - accuracy: 0.8588 - val_loss: 0.3578 - val_accuracy: 0.8533\n",
    "Epoch 88/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3355 - accuracy: 0.8606 - val_loss: 0.3597 - val_accuracy: 0.8533\n",
    "Epoch 89/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3353 - accuracy: 0.8591 - val_loss: 0.3596 - val_accuracy: 0.8558\n",
    "Epoch 90/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3352 - accuracy: 0.8588 - val_loss: 0.3594 - val_accuracy: 0.8583\n",
    "Epoch 91/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3346 - accuracy: 0.8569 - val_loss: 0.3605 - val_accuracy: 0.8525\n",
    "Epoch 92/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3344 - accuracy: 0.8601 - val_loss: 0.3635 - val_accuracy: 0.8550\n",
    "Epoch 93/100\n",
    "680/680 [==============================] - 1s 2ms/step - loss: 0.3344 - accuracy: 0.8622 - val_loss: 0.3595 - val_accuracy: 0.8475\n",
         "Epoch 94/100\n",
         "680/680 [==============================] - 2s 2ms/step - loss: 0.3342 - accuracy: 0.8597 - val_loss: 0.3597 - val_accuracy: 0.8583\n",
         "Epoch 95/100\n",
         "680/680 [==============================] - 1s 2ms/step - loss: 0.3338 - accuracy: 0.8597 - val_loss: 0.3609 - val_accuracy: 0.8533\n",
         "Epoch 96/100\n",
         "680/680 [==============================] - 1s 2ms/step - loss: 0.3341 - accuracy: 0.8587 - val_loss: 0.3578 - val_accuracy: 0.8533\n",
         "Epoch 97/100\n",
         "680/680 [==============================] - 1s 2ms/step - loss: 0.3340 - accuracy: 0.8612 - val_loss: 0.3600 - val_accuracy: 0.8558\n",
         "Epoch 98/100\n",
         "680/680 [==============================] - 1s 2ms/step - loss: 0.3335 - accuracy: 0.8596 - val_loss: 0.3574 - val_accuracy: 0.8542\n",
         "Epoch 99/100\n",
         "680/680 [==============================] - 1s 2ms/step - loss: 0.3333 - accuracy: 0.8609 - val_loss: 0.3586 - val_accuracy: 0.8500\n",
         "Epoch 100/100\n",
         "680/680 [==============================] - 1s 2ms/step - loss: 0.3334 - accuracy: 0.8596 - val_loss: 0.3615 - val_accuracy: 0.8558\n"
        ]
       }
      ],
      "source": [
       "history=model.fit(X_train, y_train, batch_size = 10, epochs = 100,validation_split=0.15)"
      ]
     },
     {
      "cell_type": "code",
      "execution_count": 25,
      "id": "9a7e12ed",
      "metadata": {
       "execution": {
        "iopub.execute_input": "2021-12-21T16:53:53.563792Z",
        "iopub.status.busy": "2021-12-21T16:53:53.547481Z",
        "iopub.status.idle": "2021-12-21T16:53:53.878163Z",
        "shell.execute_reply": "2021-12-21T16:53:53.877637Z",
        "shell.execute_reply.started": "2021-12-21T16:24:32.156781Z"
       },
       "papermill": {
        "duration": 1.080701,
        "end_time": "2021-12-21T16:53:53.878324",
        "exception": false,
        "start_time": "2021-12-21T16:53:52.797623",
        "status": "completed"
       },
       "tags": []
      },
      "outputs": [
       {
        "data": {
         "image/png":
    "iVBORw0KGgoAAAANSUhEUgAAAskAAAF1CAYAAAAAa1Xd+AAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90b
    GliIHZlcnNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXBIWXMAAAsTAAALEwE
    AmpwYAACkM0lEQVR4nOzdd3jUVdbA8e+dSe8JKRAChA6BUENHmg0FxYaKoqKubW3r6qrruq6ty1
    ty64NO4uCgCIIgtJr6IQWAiEkJCGkz9z3jztDJiGdJJNMzud58mTmV88MOhzunHuu0lojhBBCCCGEKGNxdwB
    CCCGEEEI0N5IkCyGEEEIIUYEkyUIIIIIYQQQlQgSbIQQgghhBAVSJIshBBCCCFEBZIkCyGEEEIIUYEkyS2QUup7pd
    QNDX2sOymI9milzmmE6y6ozP9Fg5IkuYk4/ody/tiVUvluz2+vzbW01hdprd9p6GObgtb6Gq0Ol1XPu9fFg5IkuYk4/ody/tiV
    UgUuz6+ty7W01hdorf/T0Mc2R0qph5VSSyvZHqmUK9a3ttbTWH2utz2uguMp9yGutD2itg7TWtoa4fiX3
    U0qpFKXU1sa4vhCi4cjnff3I5z0opbRSqltDX1fUjyTJTcTxP5Tsc5LLtY+dxSikv90XZLH0Ekda6w/W
    pgs9Z6ixticocxQDTQRSk1pClvLP9NClE38nlfb/J5L5oVSZLdTCk1TimVqpR6SCl1BPi3UipUipDKXc8Tj
    O5RzXr5RmKKV+UUq95Dh2r1Lqgnoe21kptUIp9aNS6g2l1EdVxF2bXJGcJ9Si13XG+hUirSZf91Sqn9Sql
    MpdRfqnp/tNapwE/AdRV2XQ/8t6Y4KsQ8Qyn1i8vzc5VS25VSOUqp1wHHlIrSIK6XCHPs+BDo
```

```
C3zpGhh5USsU7RgC8HMfEKqXmKKWylFK7lVK3uFz7caXU50qp/zrem2SlVFJV74HDDcD/gHmOx66vq49S6
gfHvdKVUo84tluVUo8opfY47rNWKdWhYqyOYyv+d7JcKfWKUioTeLy698NxTgel1NeOP4dMpdTrSikfR0yJLs
dFK6XylVJRNbxeITyOfN7L530tP+8rez2hjmtkON7LR5VSFse+bkqpnx2v7ZhS6jPHduX4HD+qlMpVSm1WdRi
NF5IkNxdtgQigE3Ar5s/l347nHYEC4PVqzh8G7AAigReA95VSqh7HfgKsAtoAj3P6B5Wr2sR4DXAjZgTUB3gA
QCmVALzluH6s436VftA5/Mc1FqVUT2CAI966vlfOa0QCXwOPYt6LPcAo10OAZx3x9QY6YN4TtNbXUX506IV
KbjELSHWcfwXwjFJqgsv+ix3HhAFzqotZKRXguMbHjp+rlVI+jn3BwI/AfMe9ugGLHKf+EZgGXAiEADcB+dW9L
y6GASlADPA01bwfytTlfQfsB+KB9sAsrXWx4zVOd7nuNGCR1jqjlnEI4Wnk814+72uMuRL/BEKBLsBYzD8cbn
TsexJYCIRj3tt/Orafh/kWsofj3CuBzHrcu/XSWstPE/8A+4BzHI/HAcWAXzXHDwCOuzxfAvzO8XgGsNtlXwCgg
bZ1ORbzgVMKBLjs/wj4qJavqbIYH3V5/ntgvuPxY5gkyrkv0PEenFPFtQOAXGCk4/nTwP/q+V794nh8PfCby3E
K8yH3uyquewmwvrI/Q8fzeMd76YX5gLUBwS77nwVmOh4/Dvzosi8BKKjmvZ0OZDiu7QfkAJc69k1zjavCeT
uAKZVsPxVrNe/TgRr+vE+9H8AIZ3yVHDcM8xeMcjxfA1zZ2P+PyY/8NJcf5PNePu/r9nmvgW4Vtlkd71mCy7b
bgCWOx/8F3gHiKpw3AdgJDAcs7v5/oSX+yEhy85ChtS50PlFKBSil/uX4SiUXWAqEqapn0h5xPxPtBaO0cKg+p4b
CyQ5bIN4GBVAdcyxiMuj/NdYop1vbbW+iTV/OvWEdMXwPWOUZBrMR8K9XmvnCrGoF2fK6VilFKzlFKHHHN
f9CDMCURvO9/KEy7b9mBFWp4rvjZ+quj7xBuBzrXWp47+TrygrueiAGRWpTHX7alLuz76G96MDsF9rXVrxIl
rrlZjXN04p1Qsz0j2nnjEJ4Qnk814+76v7vK9MJODtuG5l93gQk/ivcpRz3ASgtf4JM2r9BnBUKfWOUiqkDvdt9
SRJbh50hef3Az2BYVrrEMzXJeBSQ9UIDgMRjq/2nTpUc/yZxHjY9dqOe7ap4Zz/YL4qOhcIBr49wzgqxqAo/3q
fwfy5JDquO73CNSv+mblKw7yXwS7bOgKHaojpNMrU200ApiuljihTx3gFcKHjK8SDmK/fKnMQ6FrJ9pOO365
/1m0rHFPx9VX3fhwEOlbzof8fx/HXAV+6JghCtELyeS+f93V1DCjBlJmcdg+t9RGt9S1a61jMCPObytEhQ2v9m
tZ6MGYEuwfwpwaMy+NJktw8BWNqrbKVUhHA3xr7Zivwh9XZsLVCOCiRorxS2CyUmq0o7b2CWr+b3E
ZkI35SslZ73omccwF+iilLnMkd/dQPlEMBvKAHKVUe07/YEmniuRUa30QWAE8q5TyU0r1A27GjE7U1XWYr8
ucdXkDMB90qZhSi++AdkqpPyilfJVSwUqpYY5z3wOeVEp1d0zg6KeUaqNNPfAhTOJtdYw6w6VJZMu6ru/ViF+U
voOaVUoOM1u9b7fQRcivmL57/1eA+E8GTyeX+61vp57+TjuJafUsrPse1z4GnHZ3wnzJyTjwCUUlNV2QTG4
5ik3q6UGqKUGqaU8sYMjhQC9jOIq9WRJLl5ehXwx/zr8TfMpKymcC2mvjQTeAr4DCiq4thXgXeMWutk4E7
MRIzDmP+pU2s4R2MSrE6UT7TqFYfW+hgwFXgO83b4A8tdDtWWF/YBCm/ncuZtKHq2eBR5VS2UqpByq5xTR
M3Voa8A3wN631j7WJrYIbgDcdIwWnfoC3gRscX/Gdi/kL7giwCxjvOPdlzAfrQkyN3/uY9wrgFsxfBJlAH8yHfH
WqfD+06RV6EaaU4gDmz/Iql/0HgXWYD+5ldX8LhPBoryKf9xXPaa2f907JmH8MOH9uBO7GJLopwC+Y9/M
Dx/FDgJVKqTxMOdu9WusUzITtdzHv+X7Ma3/xDOJqdZyTaYQ4jTJtZLZrrRt9ZEN4NqXdCihDidf
N4LcToZSRanOL6a6aqUsiilJgJTgNluDku0cEqpeOAyzEi2EKIZkM97IWomrOmq/0IV20xXzO1wwXddofWer17Qxl
tmVLqSeA+4Fmt9V53WogZRbCCGEh3GMDP4D01/1Pa31cxX2z8DUJjpn4L+utX7Pse8GzKILA
E9prf/TJEELIUQzI0myEEJ4EEfP2J2YCZ2pwGpgmtZ6q8sxM4AkrfVdFc6x6WAHQ9SMJMtFwLDNZaH2+a6IUQo
vmQmmQhhPAsQzErraXosqXBp9Ty3POBH7TWWY7E+AdgYiPFKYQQzVqzq0mOjIzU8fU8fHx7g5DCCHqZe3atc
e01lFuDKE95VdPS8UsD17R5UqpMZhR5/scrfoqO7d9JeeilLoVuBUgMDBwcK9evRogdCGEaFrVfWY5P
j6eNWvuWuDsMIYSoF6XUfnfHUAtfAZ9qrYuUUrcB/wHOqct9tNbvAO8AJCYmavncEkJ4ouo+s5pdEi6EEC3c6ZpQo+XTuRHgBlXvuBRfKUPcWdL0uKSf8BsVgY6UqIsnGPXZnz/w8XzdNhazQ4DA06q6v1KBeWlGNP4ucLtYD04kB04fGm5AurKEl7LOeDFTUCtdHH8wxXAx
AeUqpcKVUOHCeY5sQQgghhBRkfKYdEzuJkAGdWpN6G3lCCzUnRsbIQDa6aY3aCNsZjvhiQjGR0vWPuTvljnhpQKNx3hMhPW0NP3cEKdrmsAkJt6KZSWMnzcEKsyImL1N8KhMUsP+xnd0HCgu2bUpMRVSsPxUyXDeUK1R5WBmdiP9nZnNSR54gKRLVSHxl
5eluvdlzuCa11VpO/CCGEaAYkSRZCCA+jtZ4Hx2cJNWbsqFK0YKUDrHFQA+rDwaPsxtUyhMZw4CsHk6c5e+QdbBjvcAPKhRSZFrKDjpIX6GoWD2lwadTUgAc+8sX7EHXeGE2r9Tk+fnNdGeZ
goLC2nTpo0kyC2IUoo2bdrUefRfkmQhhPAsQzErraXosqXBp9Ty3POBH7TWWY7E+AdgYiPFKYQQzVqzq0mOjIzU8fHx7g5DCCHqZe3atc
3bn3peXFxc7blr1qzhnnnvuqfEeI0eObJBYlyxZwuTJkxvkWu4gNclCCCGEEC1EmzZt2LBhAwAzZ2dN2LBhAwCPP/44QUFBB05BPPD
AA6f2l5aW4uVVeXqXlJREUlJSjfdYsWJFg8Ta0slIshBCCCFECzFjxgxuv/12hg0bxoMPPsiqVasYMWIEAwcOZOzEmPPhhBbZMfI
TIkezYsQMoP7L7+OOPc9NNNzFu3Di6dOnCa6+9dvKcoKCgk8ePGzeOyy+/nF69ejFt2jS0NstV58yZQ69evRg8eDC33303F198cb0w0dWEXhcy8wh2WhXGsw0dXNUf79RS1oLDCTIO3vmDfn7jlO1rm37Rbst+4Y6ZeIc8dsnO1Be/xeq2aYZz7PfbCxjYeOsSb9Obdjk5M8hqkaKsIcuvQ/mm4zmhMuPMeeuUYL9OhKu5xMjA81YqnAU7nV2Gx1OfoUvA4nU4C8RfiLebYzqHnZ1akQs+nVLgHRVbjufm53h0L4jy2oZlvM7Z5zVOWMHe9yIGP/TkTlYGa+WHRGjyyTmPJJuW7XODXA2
s0tHFddisTIw/jhcSWpXSwtAB+9hfn7zozF+4zQHmHy8HaXqqHMoAnn2W8uBTsjrqJeEnk/OzzutGVxvB7tMyOMIWIGWQYYT6/1LB9Ryb28r0jtkgxqQbk6dOFoyjtJpeuMMazaHWlhiYhXY3Py2lwhM
```

LY2HHnqItWvXEh4eznnnncfs2bPp0KEDhw4dYsuWLQBkZ2cD8Nxzz7F37158fX1PbWsqMpIshGjxikpt7D56
wt1hiFpKzy3kp+3p5BeXujsUITzG1KlTsVqtAOTk5DB16lT69u3LfffdR3JycqXnTJo0CV9fXyIjI4mOjiY9Pf20Y4Y
OHUpcXBwWi4UBAwawb98+tm/fTpcuXU61U6tLkrx69WrGjRtHVFQUXl5eXHvttSxdupQuXbqQkpLC3Xffzf
z58wkJCQGgX79+XHvttXz00UdVlpE0FhlJFkK0eK/8sIv3lqWw/OEJxIT4uTscUYMVe45x32cbWfzAODpHyl9D
ouWqz4hvYwkMDDz1+K9//Svjx4/nm2++Yd++fYwbN67Sc3c3x9fU89tlqtlJae/g/X2hzTEMLDw9m4cSMLFizg7
bff5vPPP+eDDz5g7ty5LF26lG+//Zann36azZs3N1myLCPJQohGo7Xm9Z928cfPNlBYYmuEexSW2Phs9QFK7
Zp5mw83yj1Ew/JxjHYVl9rdHIkQniknJ4f27dsDMHPmzAa/fs+ePUlJSWHfvn0AfPbZZ7U6b/To0cyaNYs8
Mm83Gp59+ytixYzl27Bh2u53LL7+cp556inXr1mG32zl48CDjx4/n/n+eefJycnh7y8vAZ/PVWRf8ILIRpFUamNh
77cxOwNaQBkF5Tw9vTB+Hg17L/Nv99ymOP5JQT5ejF302FuHFX71ZSEezj/G5AkWYjG8eCDD3LDDTfw1FN
PMWnSpAa/vr+/P2+SYTJ04kMDCQIUOGVHnsokWLiIuLO/X8iy++4LnnnnmP8+PForZk0aRJTpkxh48aN3Hj
jjdjt5nPh2WefxWazMX36dE3lDLlsblISkrSa9ascXcYQogzkFNQwm0fruG3lCz+dH5
PQv29eXT2FiYltuO1aQOxWhquEf/lb60gM6+IywbF8fIPO/n1zxNF+pf5fF7j53kgS828upVA+gQEdBgcTgpp
dZqrWvuseRB6vq5vWTHUWb8ezVf3TGSwZ3CGzEyIRretm3b6N27t7vDcLu8vDyCgoLQWnPPnXfSvXt37rvv
PneHVa3K/uyq+8yu1ZCOUmqiUmqHUmq3Uurh6Uuv6LKvn1LqV6VUslJqs1JKCgaF8GB
5RaXc8MEq1u4/zqtXDeDO8d2YPrwTf7mwN//3MM863Hc5l7f7jXDusExf1jwVg7qbqSy5mrT
7A2v3HefmHneW2v/zDTh7+ahPNbeDAE8lIshAt37vvvsuAAQPo06cPOTk53Hbbbe4OqcHVmCQrpazAG8A
FQAIwTSmVUOGwR4HPtdYDgauBNx3negEfA7drrfsA44CSBoteCNGsFBTbuHnmajamZvPPqwdyyYD2p1p/bdM
qYL953Tg6/WpfLYnC2VJqM/bk3non/+Qk5+7T4mPl65Hx8vC1cMjqNzZCB9YkOYW01dstaauZsO42VRzN5
wiJ3ppiPGr3syeW3RLmatPsjsDYfq+KpFXfk6k2SbJMlCtFT33XcfGzZsYOvWrXz88ccEBDT8N3PuVpuR5KHAb
q11ita6GJgFFTTKlwjAZCHI9DgTTH4/OATVrrjQBa60ytdePM3hFCuIXWmrTsAhZvP8qtH65h4xXf7yP+f3aXv
asfec3Y3bxnTho98O8Oz328slyna75rn529l8KIdXF+087dyK8opK+WbdIsb3a0d4oA8Ak/q1Y/2BbFKP5d1d6zs
bUHFKPF/DQxF4E+Xjx94U7KCyx8cg3m+kYEcADmE8+d02sk5Wv7SrODPeVvNXT4mMJAshmrHaJMntgY
Muz1Md21w9DkxXSqUC84C7Hdt7AFoptUAptU4p9eAZxiuEaEK/pWQy4e9LOJZXVOn+rJPFnP3yw4x87idun
LmaFXsyefbSRSKYMqPgRYSilePiCXlw3vBPvLE3hg+X7Tu1buDWd3Ufz6BoVyIe/7m30/OpnMC/ZcZSTxTaunt
Lx1LbJiabkoqouF3M3peFtVVyZ1IHfndWFBcnp3PPpevYeO8kzly/OX9OFFYwlPfba323uLM+MhIshCiBWi
oaebTgJla6zjgQuBDpZQF0z1tJNCt4fNHCt4/elSqmzK56slLpVKbxVBGbVmIyOjgUISQpyp+VuOkJXXHDxtn9aaB7
/cRGpWAX+7KIEvbh/Bur+ey9VDO1ZypTSmW76dv/tQmgE9vGY6/t5Wn5lafq
C7fnUmwrxeDOoad2taxTQD94kJ5Zt52Ov95Lp3/PJdHvtmM1vpUqcVZ3aMlDfDmptHxhAd4s3Br
OpcPimN
090h6tg3m9rFd+Xr9IZbulM+ixuJjlZpkIUTzV5sk+RDQweV5nGObq5uBzwmsG01r8CfkAkZtR5qdb6mNY6HzPK
PKjiDbTW72itk7TWSVFRUXV/FUKIRrFmfxYAn6w8w8gK3CZLuPVx7gx23pPDixJzeO6syQ+AhC/b0ru8xpLBbFC
5f3IyLQh3tnreeHrelsSs3hjrFdiQ7x456zu7NkRwaLdxyt8bh23pPDixJzeO6syQ+AhC/b0ru8xpLBbFC
0tTWHcgm7ScQib3awdAsJ83f76gN73aBvOXSWWzne8c340uUYF8v0V6LjcWmbgnhGgJapMkrwa6K6U6K
6V8MBPz5lQ45gBwNoBSqjcmSc4AFgCJSqkAxyS+sYB8jylEC3CyqRJRth0/QMyaY1OMF5UZWdx89vVNztzK
mRxQ31bMvcXigDy9fOYCUYye565P1tA3x49JBpkzjhpHxdI4M5J5P1vPwV5tYsftYuSQ99Xg++zPzGdk18rTrJ
saF8sfzevLH83ry6lUDmJTYjufmb+fpuVvxsVo4JyHm1LFXDunA/D+MIcJR0wzg523l89tG8MylifV6XaJmp0aS
pdxCiDobP348CxYsKLft1Vdf5Y477477qjynHHjxrX3nj17Nlu3lqVxjz32GD/++2GD/++
GMdoq/ckiVLmDx58hlfp6HVmCRrrUuBuzAJ7zZMF4tkpdQTSqmLHYfd9yilNoIfArM0MZx4GVMor0BWKe
1ntsIr0MI0cA2HszGZtc8cH5PooJ9+ei3/QBk5hVx+0frCPTx4qWp/bCcQQc/jUd0iuXVMF4ptdm4d0wVfL7MS
m4+XhXeuG8w5CTF8uzGNa95byaOzt5w6b6b8WezFPnV0cpxYtT+9G7bQjrDmQzpkcUIX41j3ZHBvmiVMP1c
hbIyUiyEPU3bdo0Zs2aVaW7brFmzmDZtWtWq3OnzdvXr0U3BOihViz/zhKwdDOEVw9pAM/7TjK1rRcrv9
9dOObY9prec4Hm/VWo/SWvfXWg/QWi90OfcjrXUfrXVM3BOihViz/zhKwdDOEVw9pAM/7TjK1rRcrv9
gFQez8nnj2kFEB5952/MHzuvJv2cncpt7x4TzCtXDWDNh0hOIQArdh8jMsiHHjFfp18w
cw+krT/1NMDHi3dvGSJwp3BuGh1/xvGKMycT94SovyuwuuuuIK5c+dSXHzmk+t8eoVgy68Ozbt4+0tDTOOuss7t27
w9/+9vfKj0/Pj6eY8eOAfDUU0/Rs2dPRo8ezY4dO2p1/SlTpjB37lwKCwsB2LdvH2lpaZx11lnccccdJCUl0adP
H/72t79Xev7LL798mlu3bk7/3FcfHfvXe3+++/h8cee+y8tK/84nKwkCVKS0vZtm0bmzdvpri4+LTrzZ4+nAzmzZ
sX4Yu8+Oz2rsA3t7YaHwNgoWfvtUM4OCJJKTLhwGbd4wxqWr8fVBSxOgZjYt6FfTcsuCkZ4F/1XfQs8+IHwAm5Qt6G
5bvaz0+XqfG5xPn1stpVfLu9kKAKBgDR8HVydUFXKuBq+HJpzgYbdBpcHw+FVgr8T5N8Vq33+ALmTPAf8WoOjb2
VW2HxVNlbvalKddEdlL3Ovbt47b6q4i3T0nbLbEvqSpaUE7o7wfmUGbyaMp8OFuBGSpOtSKQS5zFYWYz9xPTt9I7IFKsIX
f0ovLZm+rRfJbUKvuhlCFx6SnUb2oxnQzbZydpwLFYT/Hb8PlmbfidgtkSfCnDcVwfVmRXecCGAqi9PqsRFhIPHj
6CvjaIsbbaqJz4ZoJ2C03MRQxXyhHYLkBxxcT1L3E6qDJtLvhQuzqCE6BrgLDVxEmDk1Km5f/Dn4NVSBz5FmmhdlFP
O0x5tEOptaS3uL/9jcXD6Le2eECT6Nn8Vcv6GTtpMT3ebgQqLdlJQqx4lVrxTHE0l/t5uCnAkszlwzVU4elw5k2JHuf
y+M7cRWm/lqxF1MFDGh20/DmwIAgC6bg5qawW0qb6lqrTaGXRgC4k2r9cxPmfHvqkfnlSRuG/0EBl9H0NWehwZZz8SJL
MpmFGbk/7ONaa3iHaVUhmCvsSozT2MKbZtVxzTC9u7dZ/jl8SSHTsBhx8hHmF+RAfv4SzFGelp9HNTAhNLlfL6d0Hh3MvE/
Ahjcd29cyRpbqdOiTZj1+5YmWdY
Ez+5k+zv4YY0vXAd6xGrHhNafBqJdwDeSLeZAbEnjRIMKvn8WFC2bx2RcRp
1wZLb/Ga7FGAp2xbB4t3zjNSX5P5zzpNAUcsSZw+JZaQgIfgUQ8Ei6QKcnxiSdQ+jbxpQtm/qPl/mPR9HyRi7aBlcK
U5d9VaDyC6Tun+PEhcsNQpfnAQhfTmEu33wE9GcvVK+4UfKDHCdpCqFe3Fdol+hoTgyaAoxpa2aZRuORvdXoFh+0
pD4T2M5gFFw9RtjAvTyk4AsSN+tIvl+fMp0/enpxdVsRKmOHyuiMkR6nV0J6EVhIM6QftHGUd9e5sCwkvqlX3mT4+UH9
Gsi/ZaE5gCOv68dPy5Y7KOkdIuvS5+ZL7/vN7VXa0Rbmpw2+aEQtRPBH0J3B5E/71A1KXDzmJwf7mRmNn87t3H9Q9
pMH01A/V6hq4NNuh+xXPuaCS4kX2fXYYH9j2kpaJHKxJ+jGMdfQ3o2jJQVWxWlbVWSZUkaPWo/i8cOE5+pv5ASK8fMwkMN
rntaGk6/hKxQvr4jKIE6Dr2vPxXVmx4aUPvGyuW/QnvJxIwCWzeqtlfNe99xFfkcDhL/PY9sanuMqDumqOQ5UJqUXKIV/fH
yMBVRDUcjrXMOSO8NlYJE9I23S0XOKbpYKz1M3hfLl9cWGmDHbQcNFUdIW7TsnczVq23E3qVvCr6iOXQbvUDYzJzEgy
7nzJDjMnr1tVc1f0rTzOPxSHZSrmtUqoFFymJyB3uwXC+WBkFsEKBmONPdw/2fWgYuHnwbcb2qbPIz8dz1Y9c+zNW

AAezZs4cZM2bw5ZdfAmZlvYEDB5KYmMhNN91EUVHRqfv97W9/Y9CgQSQmJrJ9+/Zav9ZPP/2UxMRE+vbt
y0MPPQSAzWZjxowZ9O3bl8TERF555RUAXnvtNRISEujXrx9XX311Hd/Vysmy1EK0ArvST/DZ6oNcPbQD3aK
DKz2msMSGr5fl1Ajq2v3H6REdTKi/N1cP7cgbi3dz+VsrKLXbeff6JIZ3adMgsXlbLYzvFV3lfn+Ljd8fepge1m789
9eR/On8nizfk8mIrpGVj/Yu+j9I2wB/2HRqU/swf766Y2SDxCvOnLSAEx7j+4fhyOaGvWbbRLjguSp3R0REMH
ToUL7//numTJnCrFmzuPLKK1FK8fTTTxMREYHNZuPss89m06ZN9OvXr9LrrF27llmzZrFhwwZKS0sZNGgQgw
cPBuCyyy7jlltuAeDRRx/l/fff5+677+biiy9m8uTJXHHFFeWuVVVhYIwZM1i0aBE9evTg+uuv56233uIPf/gDAJG
Rkaxbt44333yTl156ciffee6/GtyEtLY2HHnqItWvXEh4eznnnncfs2bPp0KEDhw4dYssW8+2is3TkueeY+/evfj6
+lZaTllfDdXdQgjRgLTWvLF4N6v3ZdXq+MISG4UllbcgX7HnGJe9tYL3ftnLea8s5s7PN3Iwq3wf4YwTRYx+/iee
n29GEux2zboDxxkcb5YMbh/mz9m9Yi22fnntEGM61l1UtvgUpZgLcqmn18Gn6w6wJZDuWScKGJU1yqS9L
yjkL0fik40XYyiTrwsCqVkJFmI+nItuXAttfj8888ZNGgQAwcOJDk5uVxpREXLli3j0ksvJSAggJCQEC6++OJT+7Zs2
cJZZ51FYmIiH3/8McnJydXGs2PHDjp37kyPHj0AuOGG1i6dOmp/ZdddhkAgwcPZt++fbV6jatXr2bcuHFERU
Xh5eXFtddey9KlS+nSpQspKSncfdzJ8/n5AQs0xHv379uPbaa/noo4/w8mqYMWAZSRaikWXnF+PnbcXP21r
rc95csocXF+xgbI8o/nPT0BqPv/Hfq9l19AQvXNGPCb3MxDStNV+vO8TDX28ivk0gn/xuAHM2HuI/v+7nx23p
zLv3LNqH+QPw94U7OJZXzAe/7OW6EZ04UVjCicJSBncMP3WPl67oz+HcAnq1DSm7sdaw+BnoORHaD6716
6uT5K8BiLEdJje/iIe/NiPElU3aAyDf1CuTsQPikhonJnFGlFL4WC1SkyxavmpGfBvTlClTuO+++1i3bh35+fkMHjy
YvXv38tJLL7F69WrCw8OZMWMGhYWFW9br+WCq1UlpaekbXCg8P
Z+PGjSxYsIC3336bzz//mk2b958xsmyjCQL0Yjyiko595WlPFmHxSl+2JrOSwt
3EOBjZeXeTIpKy0aItdanrSa3NS2XX1MyKSq1c9PMNfz56828vHAH419awv1fbCSpUwRf3jGSxLhQ/jIpge/vP
YtSm537Zm3AZtckp+Xw2ZqDp1qjvfbjLtbuPw5AUnxZkhwa4F0+QQY4vheWvgDf/sEkzA2ttAi2zwWfYCz2Yi
a0KyY5LZe4cH86tqliCVRnknx0W8PHIxqMj9UiI8lC1FNQUBDjx4/npptuOjWKnUBgIKGhoaSnp/P9999
e40xY8Ywe/ZsCgoKOHHiBN9+++2pfSdOnKBdu3aUlJTw8ccfn9oeHBzMiROnf0vP9oeHBzMiROnf0vXs2dP9u3bx+7duwH48M
MPGTt27Bm9xqFDh/Lzzz9z7NgxbDYn376KWPHjuXYsWPY7XYuv/xynnrqKdatW4fdbufgwYOMHz+e559/
npycHPLyql+QqjYkSRaiEX3wy14yThSxIPkdnvNSeTO9BP8YdZ6+saG8tLU/hSW2Fm77/ip/XM2pjH6+cUs33
3s1LaPV+7H18vCoj+O5bYxXZi1+gCvL95N+3B/Xri8H/+5aWi5/sVdo4J4YkpfVu3L4s3Fu3ni262E+Xvz9CWJX
DOsI1+uS+WbdYeIDPKlY0QViajTvl/M7yObTDLb0Pb8BEW5MORmAG7sZZKqUVWNItttUOB4vzJqPzlEND0f
LxlJFuJMTJs2jY0bN55Kkvv378/AgQPp1asX11xzDaNGjar2/EGDBnHVVVfRv39/LrjgAoYMGXJq35NPPsmwY8MQ
cMYNWoUvXr1OrX96quav5sUXX2TgwIHs2bPn1HY/Pz/+/e9/M3XqVBITE7FYLNx+++ej2LFi0iLi7u1M++fft
47rnnGD9+PP3792fw4MFMmTKFQ4cOMW7cOAYMGMD06dN59tlnsdTE8+ncTERAYOHMg999xT7w4er
pRujNGfM5CUlKSdvfyEaMmy84s56/nF+HhZy
YWHLzAfUte9v5Jlu47RMyaYufeMprDUzrCnf2Ri33b8/cr+AOzPPIm/t5XokKo7T2ituWfWBr7dmAbAk5f05b
rhncg4UcSYFxZTUGJjYp+2vH1dDSUUX90CKYvBLxS8/OC2ZWBpwH97f3UL7F7f3UL7FoIty2Ff/SjdOKLPHRgGNeN6
MSADmGnH5+fBS84+jZ3PRuu+7rhYqklpdRarXWrqvOoz+f2iGcXMbpbJC9O7d9IUQnROLZt20bv3r1rPlA0O
5X92VX3mS0jyUI0kn8tTSGvuJS3pg/galH8uDW92uPrqvpg5WaQtuinPo3Wgc1gjuEs22UW8UjPLWT5
7mP07xDGjvQTfLr6IL6ILPXH+JksY3pw8uWgu7UJrDaBBlMTehTl/QlLtyfXm2DmTbELKoZFezLjaPiARjcqeqEHjD
lFfuWQfxoGPMgpG+B7d9Wf05dlBTAjnnQ+yII6wjeAXhl7+XvV/avPEGGslILLz8ZSW7mfLyk3EII0bxJkixEIzh6
opCZy/dxcf9YhnaOYEh8B8OIu2Vb3E8u6jebyxeDcX9Y8t1w5tTI8oktNyOZZZXxLcb07BrePnK/gzrHRHCHHLC3cwc8
U++sSGVJ00ViPU35vv7z2LL+8YWW5p59vGduXKpDgm929X/gRbafmOEZl74MRhiD8LEq+ANt1h8bOQsRO
O7Sr/U5hb5/jYvQiK86DvZaAURHQx96yOM0mOGwK5h6Awp/LjCo5Xvt0pYwcUZNc55OZCKTVRKbVDKbV
bKfVwNcddrpTSSqkx3NvpdR/lFKblVLblFJ/bqwYva0WiRJFkI0Y5IkC3EGtNas3pdFxbKlf/2cQrHNzn3nmHY
45/SOYUf6idNar4Fpt/bI15vx97Hy2OSEcvvO6m5qb5fvPsbsDYfoFxdK16ggHrsogeyCEnYfzWP68E71Xh0u2
M+bIN/ys39D/b154Yr+tAv1L3/wj3+D14dC8UnzfN8y87vzGLBYYYdzDkLEN3hgCryeV/3nv7LoHl/w1BLSB+D
HmeUQXyKplkhw/2vw+WslocsoSeKEL7F1W9XX+OwXmN1p+2KiUUlbgDeACIAGYppRKRqOS4YOBeYKXL5q
mAr9Y6ERgM3KaUim+MwS4BRCiuZMkWYgzsHBrOlPf+YV5yYcp9jR6wmmcaIeHKyL/4hrdPUX1+3nQbBgBnt5rzIdfR5Gb3/o1EKtsLetEK3dWsIKfbdzX4HiX3mLY2x4rfoFU1q4Pzrm6JzGPAGi/zMsdfrqcV0QAVHd/Tcr9W5rrvYNjGDv5yD6dlPCyGv9AD+OfNqjmIzPlszG7KXSf27Rq2B+anTExkd3RA2w9ExBvgBlsfKiupbwfIVcCACCHUCy8f7y7GYcxP2uW3MkFrGvM8Usyj2EW+UVKaCMCKwE8zTPdXwNR1+/jgyYPVjv0ZtufYkRCfKbxJKtLcn9i9XJtSY5wTLIhHa1b5eYmfS9CrbXKDbDfbdHz6VKbC8IQrRNuecGUZw1mHuGyrhk6Rt/3lrAe3LXOymoD9H3Tr6uE8+dH3CzhIK0pXm1IhmFbKmHqHn1+eNhMhXr4Q2QHuF6Z9C1Z87tMzEGMz18Z/f4E+jHHvnI2BhKm2x1YCYmmKIGn96oc4zddjOGEqjEnewHHAvb3n0MbttvBnFnNv64e/Gv3qx3ooXmYwYVKZy+mhHaYevgWgE4hCvYpMzeW56ZR1FXsfvDfMyehDV4MUIF5TSqr0JOiyK8AKEbTRfGkkwVo0Ro+hYgLq/QHwFQuMcfTvYJD5hq4ZOpjAtVqc3GZEZP7XMQbW7irAgSw2tJFoZb3EVWTDL4x7ay1iiEH1V6TafGtpn+GdwaRg0Y1AdXpUDDZBtdSgDD6Wi61Yt4uJuA8KRZR5BIFcm9CcuTHKp4d+z5qjSabSzO+AETVDEP73sIGmt3M8B4YbmwFsfw6LdpyhG2NrYvMW0QUHr8RHb+iyE7ug3rNMj8R7JbNu+yB28OHxwSPuRX+nFF4ksBOxGX4pMeI2UXOaQzsLEF+qyq37MgXhBxZqBvgi2NgjkQ49IyZcGoX7bj36iwsL4zAkNQ1NxQ3iEjW4C9nQlvzxGzJbl7FHWK3A3fXT0L1AqSEy0olSJYn2cYmEfVsLf6CFnbFdXnYJcPBLhdiCfhQKhVMXjcsCO39M38jmgxyn7cOAJYtNtKq1PUvVG5OaaH+LVwJc6ypgAiizTb+zvTK8GCFlImd7zbmE6zXTXvNoH77gcWYHCd0tOQbvzmMdRtwMfu8P8eeDjqdYMhQlH3P4JyHH6qn/kV+b+AxVp6Z3dBdOuanRtqXbXVmVo1pq+0+ZvPs3EVdkJnjxrTQn/L7nfJXZ8N2ve2cl6fGQlOswwqX7hnZKbyenXujAyzZWQIYTBi6blb4hi8GrlbN6Bj0XTgC7W3LYqzNRMhEuhBrRMY2B4J8ApY7cUugRBU8qC5K4+Qa4TQIPAS80nS6GUYAp2pnqsZd9BTQfHBnIvakTs7qwdMmiErzWjfYmlkqQAu/D4dtn3KDy2uXjjfDPKxsVzJ0gx4OZZw5f6ex4TwQAJ0k7hXKUiyBdE4s+BrUfEGujoEjHzTvdBiAsVMPb9MhE9uoPgVPJZfQKklcsrIejxvQ0Xwat3CYV5Y+NA+NdPRdEUZz0Mq71TCkuLqd/uwO7XKZU1QD3fLBXnhqJbBp5y1sBVc3ymEqDuB5J/EEt2Lao8=
AAezZs4cZM2bw5ZdfAmZlvYEDB5KYmMhNN91EUVHRqfv97W9/Y9CgQSQmJrJ9+/Zav9ZPP/2UxMRE+vbt

3g9y2DoLbQppt53vdymDYLLn+//M+A6XBsJ+RlVH3tiorzYcd86H0xWB1JfJuucHyfGdGuijNJ7uSYsJJRSYeLjb
NA2+GnpyrvyFF80oyQt+lS+3ibl6HAbq11ita6GJgFTKnkuCeB5wHXHlEaCFRKeQH+QDFQj68BaubjZaFIkmTR
QjW3+VyiZvX5M5MkWYgz4Owy8e3Gw6e2rTtwnGN5RUzsW1auEB8ZSNeowHIlF3sy8rjirRX8a2kKVyV1YO
rguNOub7UoRnaLZP2BbKwWxUUuCfETU/qy+IFxBPo2Qbvz3MOQuQusPrD8H6bsYq+jHtk5iq0U9LzAlF64/
gwwM685vKH299v9A5SchD6Xlm2L6Ar2Usg5YJ7b7bD7x/KJrjNJjh0A3gGnjyQ7W8oFRsHB38ykw4qyUsru1
zK1Bw66PE91bDtFKTUI6KC1rtiS5EvgJHAYOAC8pLWu3Yo2deQj5RaihfLz8yMzM1MS5RZEa01mZiZ+ftXP16
lIFhMR4gysTDH5w8KtRygq7Yuvl5UFW47gY7UwvmdUuWPP6R3DB8v38sPWdBYmH+HbTWn4eVt545pBT
OrXrrLLAzCmeyRzNx1mVLfIciPNVosq19qtUTlbvZ3/DMx7AOY9CCePQuezaj63bT9AQdp66H5u7e635WuTy
HZyaWHUxpG0ZqaY0out38CXN8GMeRDvOC4/00za8wmCqJ6njyTvXmRayl36iXkNi5+BLuPLEn0oq3tu02KT
5GoppSzAy8CMSnYPBWxALBAOLFNK/ai1TqnkOrcCtwJ07Nix4u4a+XhZyC8+s0UFhHCHuLg4UlNTyciow7dj
wu38/PyIizt9MKo6kiQLUU+ZeUXsSD/B0M4RrNqbxbKdxzi7dzTzk48wqlsbgv3KJ7Bn947hX0tTuOW/awj29
eKifrE8cH5PYmroRDGuZzSBPlauHVb3RKTB7FsGvqGQdJMpu9j4idkeX4sk2S8EIrtD2oba3av4pLnHgGvKSi2
gbGQ3aw9wjkmkwZRgnEqSs0wds1IQ1dv0WXaV/A34hUH382DMUfjuDyZx7n5O2THOuueIFltucQjo4PI8zr
HNKRjoCyxx1LK3BeYopS4GrgHma61LgKNKqeVAEnBakqy1fgd4B0wLuLoG6eNlIbtARpJFy+Pt7U3nzp3dHY
ZoAlJuIUQ9rdprRpHvP7cHof7ezN18mOS0XFKPFzCxb9vTjk/qFM5fLuzN29MHsfrRc3hxav8aE2SAmBA/Nv7
tPM7vc/o1m8y+ZdBppGOCnmNCW0j72ieS7QaYkeTa2LkASgtMVwtXQdFmhDhzj+mWsesHsz3XJf/Lz4SAC
PM4uhfkHSnrZOHaUs7qDQOuhdCOsOSZ8iUbmSkQFAO+wbWLt/lZDXRXSnVWSvkAVwNznDu11jla60itdb
zWOh74DbhYa70GU2IxAUApFQgMBxqll563VVFSKl9XCyGaL0mShajgYFY+n646wMmi6r8K/i0lE39vK4M6h
TOxT1t+2JrOtxvTsKiyiXquLBbFLWO6MLFvO/y8rXWKybVFW5PLOWTqdJ2lFXGDYehtMPSW8mUK1YkdaCb
9nThS87HJ35gkteOI8tudbeCy9sCO78FWBCjISS07xjmSDBDtaOjgrEt2tpRz1jl7+cCwW+HQWjNRzykrpSXXI6
O1LgXuAhYA24DPtdbJSqknHKPF1XkDCFJKJWOS7X9rrTc1Rpw+XlbpkyyEaNak3EIIF+sPHOd3/1lD5sli/r5wB
3eN78a0YR3x9To9qV25N4uk+HC8rRYm9WvHZ2sO8u/l+xgSH0GbIN9Krt5COeuRnW3VAC58oW7XiB1ofq
dtgJ4Tqz6u6IRZYW/Q9WbUuqI2XeHwRpNIh8SZUePctLL9+ZkQ5qg0iHIspZqxDTqNMC3l/COg89iy4zsMc8S
1HkIckyKz9tS+drqZ0lrPA+ZV2PZYFceOc3mch2kD1+ikBZwQormTkWQhHBYkH2Hau78R6OvFm9cOolt0EI9
/u5XzX1nKugPlF5/IOlnM9iMnGN7FjFqO6NqG8ABvim32Skstmq3Fz8Ivr5bflp8FH0w0nSMA9i01dbwxifW/
T9tEUJbKO1xk7YU3R8I/+sPrQ6C0sHxXC1cRXeH4ftizCPpcAqEdTi+38HeUW4TGgU8w/Pi4ufbW/0HCxeXrn
GP6mric9dJFJyAvvUWPJLcU0gJOCNHcSZIsBLB4+1Fu/2gtvdqG8PXvR3JhYjs+vWU4/71pKCU2zdS3f+XlH3a
ealm1aq9pNTa8i0nIvK2WU8nxee6sHa4Luw1+exPWf1h++8FVcOBX+HQa7FxY1urNcgYfF75BENmj8rrkJc+
ZEocOw8zCJKP+AB2GV36dNl1B28BWbBLp0PamHARM/+TC7LJyC6Xg3P+DHhPNtftdBSPuLn89nwAzwc8Zl
7P9m4d2tmhOfKxKWsAJIZo1KbcQrU6JzY6XRZ1ape7oiUIe+GIjPWOC+fSW4fj7mK/5lVKM6RHF9384i8f/l8x
ri3bx846jvHLVAH5LycLf20pi+7BT173/vJ6cl9CW9mH+ld3W/UqLTR2u0+GNph1acR6UFIK3YxKhs21aZA+Yd
Q3YS2D478/8/rEDTbcJrctqmY/tgs2fw4g74bynar6Gc4Q3tCO0H2xKQYpyzAhwiWNNDGeSDDDkZvNTU1w
755u4nO3fZCS50fl4SbmFEKJ5k5Fk0arY7Jopry/ngn8sY9vhXOx2zQNfbCKvqJTXpg08lSC7CvHz5uWrBvDGN
YPYn5XPha8tY87GNJLiw/HxKvtfKDLIl/G9opvy5dTe9nnwXEdTquDkXFZa281CIU5Ht0NwLMz4Dtr2Nds6jzn
zGGIHmlIG10lyPz9v+hqPvLd212jTDVCm1EIp02EDTF2ycyERZ3eLWsc1APKPmQmALb/9W4vh42WRiXtCiG
ZNkmTRqny/5TBbD+dyICufKa8v59YP17J0ZwaPTk6gR0z1Lb8m9WvHgj+MYUh8BFknixnRtU21xzcr6z8ybd
WSvy7btneZWZUOyq9Ml7HNtE/zD4cbvoUbv4eYhDOPwXXyHkDGDtj8JQy9FYKiqjytnMA2cMMcGPugeR7
qSJJzUl2S5Dr+uZyKa71p/xYca8owRKPytlqw2TU2u7SBE0I0T5Iki1ZDa80bi/fQJSqQn/80nnE9o/hxWzrn9I5h
ei0X6ogJ8eO/Nw3lk98N46ZRzbSZfHoyFOaUPS/MMcs8Q9kCHLYSU3fc9zKweJWVWNjtkLHT1OmC6RXca
WTDxOWcJLflSxPHgkfAJxBG3lO363QeU9bD+NRI8qH6J8kxfcx7cHiDGUmWeuQm4fwWRuqShRDNldQki1Z
jyY4Mth3O5cUr+hEV7Mu/rhvMqr1ZJMaFnqpPrg2lFCO7RTZipGfAbjedKTqPgas/Ntt2fG8muiVcAltnm7rbg
uOmFrnr2XBwddlIcvY+M+Ic3avhY/MJgNhBsOUr8wMw5k9mdLi+gttheiUfMhMRoe5Jsrc/RDsm72XugV6T
6h+PqDUfR+/volJ7nfuGCyFEU5AkWbQKWmteX7yb9mH+XDLQjD4qpRjWpQWVTNRG7iEzGW/7d6asIXaA
6Skc2sFMjNs62zxXji+R4kebhPjIZvPcmSxHN0B5RWWun13WjUJZznzU1svHrMSXm2pW0YO61ySDKbnY8rX

5h4OMJDcJGUkWQjR3Um4hPF6pzc4PW9NZu/84t43tgrc7V6+ri42flSWvteWceAamtVrBcbPSXMIUs8hGh
2EmSd63zCy2ERRtSiuy9kJxPhzdas6N6tlwr8OVb7BJyqN7QVSPyhcMqauQ9o6Je1ngHWhGhusqdqBJkEE6
WzQR50iydLgQQjRXMpIsPEpmXhH3f7GR/CLz1XteUSm7M/IoLrUTHezLlUkd3BxhLeUdhdm3m9Xhrp9d+/
OcLcwG3whr/w0/PW1auPW5zGzvcxnMfwgyvGHwDLMtuheg4dhOyNhuRp19q5/E2KyEtjd11IHRdS+1cGo
3oOyxjCQ3CedIsiTJQojmqoUMqQlRO//bkMaSHRkoBVaLIjrElxtGdOKlqf359u7RLaf2cdsc05pt71I4eaz252
WlmJZq5/6f6U6x+l1HT+FBZn/CxYAyiXPns8w25yS9jO2m3CKqEeqRG1NIXNnEvfqUWoBj8p43oCC8mU7I9
DCnkmQptxBCNFMykiyazM70E/x94Q4evqA3nSMDG+UeC5KP0DMmmM9uG9Eo128yW74xSW7Bcdj2LST
dWLvzMveYHr9+oTDyblj0RFlPYYCQWOg4Ag6sgE6jzbY2XU2CmL7FjCZ3HdcYr6jxhMSaUonjeyGsdl1KTuPla
xLl/MyyRVVEo/KWcgshRDMnI8miyXyy8gALktO57M3lrN2fVa9rzF5/iPs+20BOQclp+zLzili9L4vz+7aQZaGrc
uII7F8OQ28zi2e49jauSdaesoUwht4GSTebPsSuJvwFzn6srKuE1Rsiuzu6YBSVjSy3FM5eyZl76l9uAab38riHGy
YmUSMZSRZCNHeSJIsms2xXBv3iQgkL8GHauyv5fvPhmk+q4J8/7eKb9YeY+vYKDmUXlNv347Z07BrO7xPTU
CG7x9Y5gIY+l5qffb+YGuWa2G1wfF9ZTa1vEEx+2UzYcxU/Gs66v/y2qF6Quds8boz2b40pJM7xQJ9ZktxrEgyc
3iAhiZr5ykiyEKKZkyRZNIlD2QXsyTjJxf1j+eqOkfSNDeHuT9fz657MU8dknCjiqn/9yshnFzHy2UWMf2kJO46c
OLV/99ET7Mk4yaUD23M4p5BL31hOclrZohnztxyhQ4Q/Ce1CmvS1Nbjkr00LtuheZqKdtpsa5ZrkHDT9kOvT
nSHaZfS4pdUkO0eS4cySZNGkvKUFnBCimZMkWTSJX3ZlADCmRxQRgT7MvGkondoE8PuP13IwK5/jJ4uZ/t5
KNqXmMLJbJKO6RXIkp5D3f0k5dY0FyekAPDSxF1/dMRIvi+KGD1aTll3AicISlu/O5PyEtnVaGKTZyU0zK+H1u
dQ8j+4NkT0heXbN5zo7W9SnO4MzMQ7rZFbBa0mC2pb1fa7vxD3R5KQFnBCiuZMkuZXRWvP95sPkF5c2yP
U+XXWAS99czqbU7GqPW7rrGDEhvnSPDgIgxM+b924Ygs2uueW/a7jh36vYm3mS925I4qWp/Xlxan8uGdie
ORvTTtUfz99yhAEdwmgb6kePmGD+e/NQCkts3PrhGr7fcoRim52JLb0e2ZkMO5NkpcpKLrbOgV0/mhXyKp
Pl+AfFmYwkR7ewemQAq5dJlEFGklsQaQEnhGjuJEluZdYdyOaOj9fx3rK9Z3wtu13z+k+7WX8gm8veXME/F+
2itJKvTm12zfLdxxjdLarcKG/nyEBev2YQO9NPsDUtl7enD2KUy3LP1w7rSGGJna/XpXIou4DNh3I4v09ZEtwtO
ph/XD2A5LRcHv1mC5FBvgzqGH7Gr8utdn4P0X3MRDqnvpeb359fBx9fDu+fA3t+Ov3czD1mMY3gevxDIbw
z+EdA+6T6xe1uzpILSZJbjFPdLaTcQgjRTEkLuFbmx22mZGH2hkPcPaFbtaUJRaU2rnjrV6wWxcX9Y5ncrx3RI
WXtsX5NyeRQdgFPTunDyr1Z/P2Hnfzn1330ahtCz7bBTB/eic6RgSSn5ZCdX8KYHpGn3WNMjyjeuyEJP28rI7
uW39+3fSgDOoTx8coD2LXZVnFS3tm9Y/jT+T15Yf4Ozk2IwWJpwaUWdrtZSjpxavntUT3grtVQkA1o+PImW
PwMdBlf1toNyjpb1KfcxOoFd64Ev7D6x+9OIe2B1ZIktyC+MpIshGjmZCS5lVm0LR0fq4WUjJNsPpRT7bGfrDz
A5kM5nCgs4YnvtjLiuZ9YmHzk1P7P1xwkxM+LqUkdeP2aQbw9fRDjekaTU1DCh7/t58p//creYydZtssshuE6S
uxqQq+Y0xJkp+nDO7H7aB5vLdlNz5hgukQFnXbMHWO78upVA7j/vB61fRuap+N7oSgXYgecvi+yO3QYAh2
Gms4UqavNctOuMvdAmy71v39QNHj51P98dwp1dLiQJLnFkBZwQojmTpLkVuRgVj470/O4fVxXfKwWZq9P
q/LY/OJS3li8m+FdIvjxj2P58Y9j6RkTzF9mbyEnv4Sc/BK+33KESwa2P7WK3cS+7U6tbDf37tHY7Jpr3/2Nbze
m0Sc2hMgg3zrHPLlfO0L9vTmWV1xlazelFJcMbF+v6zcraevN79iB1R834Fqzit7ip0E7hthtpZC9v371yJ4gfjS0
6w8Blf9jSzQ/MnFPCNHcSZLswbTW2Jx1CpSVWlw2sD0TekUzZ2NapTXEADNX7ONYXjF/Or8nSim6RQfxwh
X9yDpZzHPztzFnUxrFpXamDu5Q6fndY4L5701DySsqZfuRE5zVPaper8HP28oVg80o4Xl9WvikvJqkrTdLStfUg
s3LB8b+CdLWwc4FZlv2frCX1q+zhSfoeQHcttSUjYgWQVrACSGaO0mSPdTm1BzOe2Upl721gqJSGwCLth2la
1Qg8ZGBXDKwPcfyilju0qfYKaeghLeX7GFCr2gGdyprqdW3fSg3j+7Mp6sO8vpPu+jVNpi+7avuSdy3fSgzbxpK
3/YhXDIwtt6v5Q/ndOft6YPp2z603tdoEdLWQ9tEswJeTfpPg/B4+OlJKD55Zp0thHADGUkWQjR3kiR7mFKb
ndd/2sWlby4n62QxGw9m89KCHZwoLGHl3kzO6W1KFsb3iiLEz4v/rT902jVe/2kXuYWlldb43ndODzpE+JOe
W8SVSR1q7Ek8qGM43919Fr3a1n+Bj2A/75bf2q0mdjsc3gjtBtTueKs3nP8MHN0Kn1wFRzaZ7a11JFm0ON5
W89khSbIQormSJNnDvL54Ny8t3MnEvm1ZdP9YrhveiXeX7eWZedspsWnOdiTJvl5WJvVrx/zkI2w7nAuYlm5
Pz93Ku8v2clVSB/rEnj5y6+9j5cUr+jOwYxiXDmx/2n5RT5m7oTiv5npkV70mwaXvwP7l8NPT4BMEgfUraxGiq
Sml8LFaKLbpmg8WQgg3kAK+Fiw7v5hQf+9To7k2u+bTVQcY1zOK168ZBMBfJvXmt5RMPl11gPAAbwZ1DD
t1/vUj4pm76TAXvraMi/rFUlxqZ37yEWaMjOevkxOqvO/wLm345vejGvW1tTq1nbRXUb+pYLHAV7eYUeSW
vNqgaHV8vCwykiyEaLZkJLmZOZJTWKvj/r18LwOf/IF/L993atvSXRmk5xZx9ZCyyXR+3lZemzYQH6uFCb1i8LK
W/ZH3bhfCsgcncMfYrvywNZ0FW4/w18kJPH5xH6wtud9wS5S2HrwDILIebez6Xg7XfQMTn2/4uIRoRD5eFo

ptNneHIYQQlZKR5Gbkx63p/O6/a5h54xDG9Yyu9Bi7XfP0vG28/8te/L2tvPbTLqYmxRHs580Xaw4SEejDhF7l
W6X1bhfCd/eMJjr49BZpoQHePDixFzeO6kx6bqHnT45rrtLWQ9t+9e/O0GVsw8YjRBPwtioZSRZCNFsyktyEc
gpKOJZXVOX+T1YdAODFBTuw20+v08vOL+a2j9by/i97mTEynk9vHU52fgnv/7KXrJPF/LA1nUsGtD/VpN9Vj5
hgwgKqXigiKthXEmR3sdvMxLvKFhERwoP5eFkokZpkIUQzJUlyE5m/5QjjXlzMlf/6Fa1P/0shPbeQJTuO0jMm
mOS0XOa7rGwHsGxXBue/upTF24/ymKMkYkCHMCb2act7y/Yyc8U+SmyaK4fENdVLEg3l2E4oya97PbIQVV
BKTVRK7VBK7VZKPVzNcZcrpbRSKsllWz+l1K9KqWSl1GallF9V558pH6vUJAshmi9JkhtZfnEpf/piI7d/tBaAlIyT
bHV0k3D19bpD2DW8ce0gukcH8feFOyi12SkssfH4nGSue38VwX7ezL5zFDeN7nzqvD+e14OTxaW8tmgX/eJ
Cz6jVmnCTQ+vMb0mSRQNQSlmBN4ALgARgmlLqtJm4Sqlg4F5gpcs2L+Aj4HatdR9gHFDSWLH6eFkpkiRZC
NFMSZLcyP71cwpfrkvlrvHd+P7eMVgULEhOL3eM1pov1hxkSHw43aKDuP+8HuzJOMlLC3cy+Z+/MHPFPma
MjOe7u0efVhLRIyaYSwaYVmxTB8socotzbBcsegKCY6FNN3dHIzzDUGC31jpFa10MzAKmVHLck8DzgOts4fO
ATVrrjQBa60ytdaPNrPOxKllxTwjRbEmS3MgWJB9hSHwED5zfk7ahfgyJj2DBlvKlFGv3Hyfl2EmmJpmuFOf3a
Uti+1De/nkPJwpL+O9NQ3n84j74eVsrvceDE3syfXhHLh0kSXKLkrEDZk4CbYPrvgZL5X++QtRRe+Cgy/NUx7Z
TlFKDgA5a67kVzu0BaKXUAqXUOqXUg1XdRCl1q1JqjVJqTUZGRr0ClRZwQojmTLpbNKLU4/lsP3KCRy7sdWr
b+X3a8sR3W9l77CSdIwMB+GJNKgE+ViYltgNMk/3nLk9kzsY07hjbtdoJdwDtQv156pLExnshomp5R2Hen6D
UMRgX1tG0YrO4/Ptz6YvQtj/0OK9sW/YBmDnZPL7hO4gu+29EiMaklLIALwMzKtntBYwGhgD5wCKl1Fqt9a
KKB2qt3wHeAUhKSqrX7DsfLwuFJZIkCyGaJxlJbkQ/bT8KcGopaIDzHcsrL3BMzDuSU8h3m9KY3K8dgb5l/2bp
ExvKny/oXWOCLNxsx/ewdTbkpMLxfbDqHdj2v7L9B1fBT0/Bt/dAicu32oufgaJcmCEJsmhwh4AOLs/jHNucg
oG+wBKl1D5gODDHMXkvFViqtgGDGitQb6tFi2EEM2WJMmN6MdtR+7kSGUiXqMqBT29qH+ZPYP
pQFyUcoLLFx24drALh1TBd3hSnOxOEN4BsKt/8Cd6wwi4EseR7sjr/4lzwLXv5w4jCs+4/Zdmw3bPoMhvwOo
nq6LXThsVYD3ZVSnZVSPsDVwBznTq11jtY6Umsdr7WOB34FLtZar8Emo6AoGUL5EqxlTkqMJ5b2vt2hdWu
YjTeoGJ6kTVoiKupWKhTsJUQExXKKlIWFxK7fu9U0itvXGZPb3Ub9V5zrMDhXYQAYYfrNKYBDlaI5tjfnjBbGl1yRTbbTTfJJPVRc3VJ9WvRxO7W1+rpq/vuQfM7WbqSVvWxhp0kVsj9x0m+aSUFEmNVv8dmlv4f8W2O8xyzYTAt7CH4X1VLhs66nxmRyRd59ipJ9dUKVpjVblzbRZ9eoJlsuq8aXTUXEkk/8v9vm+HzVvVVqbJ8S1+XzIfcgYoqZprNrk0YyTwNxm/gNTi18wGsPB5ZXGmtJXFx0HhwcI1SvEv3d9ZtoJkh5qZmNbNkeo3Plg0RfqtLxV7a2LCy2qf/lLPWTglXbe

HqcD1v/BxOfB2sVf5yuZRZZeypPkrU2o6POP7agGPDyqTk2V6VFpuvDiDvN8+jeYPU1SXLiFXW7VlXsNhNjVa
+1Ie1fbhZFmfQyJFzc+PcTojUJMn3kI1UORTKSLIRohmpVk6y1nqe17qG17qq1ftqx7TFHgozWeqvWepTWur
/WeoDWemEl13hca/1Sw4bvJvMfhrfPAlsJK/dmse5ANvcNC8T7lV6w4eOaz9ca3p0AMydB0Ymaj1/5FvhHw
NBbT9/X51KTyO1fXvX5WXvA19F8pKq65KUvwit94NW+5ufz62uOq6L0ZPOPB2dbNKs3tE2Ewxvrfq2qLPyri
e/o9oa7ZlWSvwbvQOh+buPfS4jWJsjMP4kiW8othBDNUiP1KfNwB36FYzsoWvsxL8zfTmSQD5ecmAVFubB
3ac3np62D9M0msf3ocijMrf749K3QcYQZOa6o+3ngHWBKLqqSuQc6jwFlqbpl3M75ENUbLn4duk4wsdnr+
BfX4Q3mt2tbtNiBpk65rteqTGmx+UfIicPmHxjpW8/8mlWxlcLWOdDzgvJLSwshGoZj1b1IlUOJzXO/eBRCtFy
SJNeV3QbHdgJwfP4zJB88xjMTwvDa8KHZX5uWZ1u+Bos3XPxPMyHvo8uqTpTtdji+F9p0qXy/TwD0mAjb5p
jEriJbCWQfgKhepl1ZViUjyYW5JpHtNQkGXQd9LjMJf216MJ88Bpu+MO3otn1r6qbDXJaZjR0AxScqv29dpSw
xq/id9zRYvOA/k6tPlIvz4eCq07dn7ITcw9Xfa99SKMgyI/VCiIbnKLeIUjkykiyEaJYkSa6r4/ugtJDFlhG0tacze9Q
+zsv8yNQWD7oBju2qfmRYa0ieDd3OhkHXw9SZkLoa1rxf+fG5h6C0ECKqmfjW7yrIz4Tv/nD6iO3x/aBtZuJc
m66Vl1sc+M0c09kxz9JZLlGbhP/Hx+Hr38HXt8Cen6DTqPJ11rGDzO/U1TVfqybJ35jOGUNvhRvnmUR5XjUr
nf/yCrx/bvlFVApz4P1zYOFfqr/Xlq/BJxi6nXPmcQshThfQBo0ySbKthjaWQgjhBpIk11WGqYX9QE/mZPRAem
9/E9Z/ZBLeXpMBDUc2VX1+6hrITS0boex9EUQnVF2m4RyBra47RI/zYcyDsP5DmHNX+b7JzvMjupqfrJTT+xb
vW2pWx+swzDyP6gVefrVLkvf+DN3OhbvXmZ8rPii/P6qXqafe90vN16pOaRFsnwu9J5sJhW26QtLNsH9F5aP
CWsOWr8zjJc+Wbf/tLZMoZ+6u+l62EjMq3utC8PY7s7iFEJWzemHzb+OoSZZyCyFE8yNJch3Z07cB0DVhMIH
n/RXyjpha39F/LKvFTdtQdkL2AVjxukm8wIyGWn1MratT/GgzmltafPoNM12S3KooBRP+AuP+bGp2v73n9P
OdI8lFuaZEwtXeZWaFPGftrdXLMeHO5XXYSmDpS+XPPb7fvL7u55Zd38u3/LUtFogfZe7h6tBaWPdh+W1am
xUEM3ac/hr3LIainPLlD30uBbTp7lHRkc3mHwgxibD7B1N2UXAcfn3D7M8+UP74ojxY8Bf47j746mZT1iGlFkI
0Kpt/JJEqRxYTEUI0S5Ik11Hugc2k6kgGdu9oJrj1vcKM4oa2h6BoCIkrPwK75Dnz1f4XM8xoaPI3ZuTVuSAIQP
xZUJJvJvRVlJVieiEH12KRknEPw4i7zMj2McdIadYeU6IQ0KYs0XatDy7INiPfFfssxw40XSmco9I7F8BPT8Kqd8q
OcY4Ox59FteLHmFX/ju8r27bwMTPqfWRL2baUJTD/IfOeVZT8tVltsMu4sm1RPSCmb+WTFpO/AWWFaZ9A
QKQZTf71TfOPhMQrTcLsWhaTstj0RE7+xoxOxw40f75CiEZjD4yWmmQhRLMlSXId2dK3ssvenmGd25gR3Cv
eh7F/KjsgdkBZklxaDNu+g/B42P4dvHcOnEg7fYQyfjSgTh9tBTMSHNHFjMjWxvA7zO+t35Sd36aLidVZsuFal3
zgV9D20xPd2IFQnFdWluBMRJO/KSvX2LfMJN9RvaqPyVnr7Hx9J46Utaz72ZEQa11WFrFzvpl051RSCNvnmd
IUq3f5a/e5BA7+Bjkui0BqbZLqLmMhrCOMutfUSy//ByRMKRvFdx1Ndr4n926EP+2GW5ecPiouhGhQOjCaS
CRJFkI0T5Ik14WtlJC8vaT7daZtaBW1qrEDzUhtQbYZnSzKgQteND9HNpnFNXpOLH9OQIQZEd1XSV1y1p6q
O1tUJjTO1BYnzy473zmCHNbRjK66jiTvXWZiihty+usAk/CXFMCO701t8bGdcHSrSUT3LjMJfk0JfFQvM5rrHH
neOgfQpovGtm/NiPWeRXBwpdlWkm8WT3HatdB0yKis/KHPZY5rupRcHN5gRq2d+4b8zrSbshXD2Ich3NF9
wzVJztpjYnQd4RdCNCoVFG1awJXKxD0hRPMjSXId2DJT8KYEa0zvqg9y1iUf3mg6JDhLBIbdaia1Tfp7+VXznD
qfZepmS4vKttltJtmrrh65Mn0ug/QtppQhJ7VsBNnqbRJE15Hkfcugw9DTJ6hF9jD9l9PWmyS15CRc+KKpv97
ytYkrN7XmUgswo9jxo829nKO80Qlw0asmKV3yHCx+1rSou+RNk9Amf23O1Rp+edm0les85vRrt+kKbfuVHQ
+OFntepqUdmDZ5U16Hic9BTEJZi7pyI8kpLXPpbCFaMBUcjb8qRtdmUSUhhGhikiTXwaGdpmY4qsuAqg9q5
xiBPbgSdswr68YA0Pdy04e4MvGjTau31DVl23IOmtHPuiZvCRcDyrRA0/bySXZE17KR5PwsM8GtskTXYjXJZ9p
6U2IREAkJl5hjk78xCS9UnrhWpvNZpp3d/uWmxKPPpSZBHnGXeZ8OrYExD5jJgwlTYOdCM5lu53wTw5g/nV
5q4dTnUtNi7ug2U2ucPBu6jDcj9E49zofht5vHAW3MPwCy95ftdx1xF0I0CUuw6ZXsVXCshiOFEKLpSZJcB0dT
NgDQKzGp6oMC25iyhpVvm0lite2Q0GkkoMqST6hdZ4vKhMSaFfqco6uuSXabrmbUVGvHJDxdVjNcUexAOL
zJTNpLmGK6XvS51CSUK/9lRnwje9QupnhHMv39Q+a3830ZdrsZbQ/rCAOuLdtXWmAS5MXPQHhn6H911d
d2XuvN4fB8vJkkWN37rpQZTXaOJBefNKv41aWsRQhxxryC2wLgU5Dh5kiEEOJ0Xu4OoCWxp2/jsIqmXVRk9
QfGDjQ1sv4R0Hls7S7uHw7t+pk633EPm23OFe/qUwbQ9zI4sMI8jnBJ/iK6mtKJ+Q+bRL7vFdBheNWvY+Vb5
vGpvs4Xw9z7TTlHn8vKLxxSncjuZoWt9C2mLVtkd7PdLwSu+9p08HCOFHccAUFtYeFfzUTHS96qehQZIKIzXP
URZB80z739IHFq9fGEdTQt7KDsfZaRZCGalCXYLE3tUyQjyUKI5kdGkiuw2TWvLdrF/C1H0C6LbtjsmvCTe8gO
6lbzRZyT3irrxlCd+LMgdZWZKAdmJNkn6NTyrXXS21Fy4R9evuzAOVq68m2zUt9l71Q98c75OoJiHCPdmJHyL
o7Ev6oR6Mo465LBdKRw1X6wqRV2sljMMSfSTOKaeGXN1+99EYz4vflJuqmsxKUqYR3LRpJde0kLIZqO47PN

```
V5JkIUQzJElyBW//vIeXf9jJ7R+tZcoby1m0LZ1d6Sf4KTmVTjqt+kl7Tp1Gmd/9rqrbzbuMMzXIO+aZ51l7zChp
bUdrXQXHmKWv2/Uvvz06wUy+6z/NjNBarFVfo003U1KROLX8cf2uNhPjuoyvW0zdzwOLtxnlrokzMR7/iCnz
aGjhnUznkYJsl1UJpdxCiCYVEIENhX9RprsjEUKI00i5hYuNB7N55YedTEpsx7ieUbz64y5u/o+ZSNdVHeJcXxvR
XQfUfKEOQ+H+HeCot6u1rhMgsif8/IKZJJe5x6x8V19T/wNUWO41JBbu22piqyn5tljg97+d3o2j35Vmwl5ILR
Y4KXfeVeYfArV5X+IGwx+31/0etRXW0fzO3m9qtINiKu86IoRoPBYrxwkloFiSZCFE8yNJssPJolLunbWemBA/n
rk0kdAAby4eEMuynccoLLXR7tBRWAlhnfrV7oJ1TZDBjNaOewi+vAk2f2kSuIqlCXXhG1T59roknoFtTt+mVP2
SV6Xq9r40VoIMLknyAelsITyOUmoi8A/ACrynta5kGUtQSl0OfAkM0VqvcdneEdgKPK61fqkxYz2uwggoyWr
MWwghRL1Ikuzw9LxtHMjKZ9atIwgNMHXEvl5Wzklw1ANvWwS+IRDVs3EDSbgUol6EBY+AvVSSt8bi7JV8fL
8Zse9xnnvjEaKBKKWswBvAuUAqsFopNUdrvbXCccHAvcDKSi7zMvB9Y8cKcNwSTnSpJMlCiOZHapKBUpud/
60/xBWD4xjaOeL0Aw5vMivDDb/D9PFtTBaL6W6R75jIInWyjcM/HHyCIT0ZTh6Vf4wITzIU2K21TtFaFwOzgC
mVHPck8DxQ6LpRKXUJsBdIbuQ4AcixhBMsSbIQohmSJBnYkpbLyWIbY3tEV37AkufANxSG/75pAup9sVmm
GqTjQmNRypRcpCwxz+V9Fp6jPXDQ5XmqY9spSqlBQAet9dwK24OAh4D/q+kmSqlblVJrlFJrMjLq3+c40yuas
NJjZvEgIYRoRiRJBn5LMZNGhnWpZBQ5bT3smAsj7gT/sKYJyGKBya/A8DshMKpp7tkahXcybeZARpJFq6GUs
mDKKe6vZPfjwCta6xozVq31O1rrJK11UlRU/T+ndvsmYMUOh9bW+xpCCNEYpCYZkyR3iw4iMsj39J1LnjMr
wjmXNG4qHYaaH9F4nJP3QMpahCc5BHRweR7n2OYUDPQFlijT4aYtMEcpdTEwDLhCKfUCEAbYlVKFWuvX
GyvYvX59zIODq8p6sAshRDPQ6pPkUpud1XuzuHRQ+9N3Ht1ulkae8Cj4hTZ9cKJxOZPk4FjwCXBvLEI0nNVA
d6VUZ0xyfDVwjXOn1joHOLVsqFJqCfCAo7vFWS7bHwfyGjNBBij1CWG/tROdDv7WmLcRQog6a/XlFs565OF
dKml15lzWuaYljkXL5OxwIfXIwoNorUuBu4AFwDbgc611slLqCcdocPi42Uh2doLDq4Gu93d4QghxCmtfiR5
pbMeuXMlSXLaetMFwZlMCc/iHEmWUgvhYbTW84B5FbY9VsWx46rY/niDB1YJH6uFTZZeXFi4ADK2l1+iXgg
h3KjVjyT/lpJJ16hAooIrqUdO2wDtBtRvWWjR/IXHg5ffma1qKIQ4Iz5eFjbg6D8vJRdCiGakVSfJpTY7q/cdr7zU
oqQQjm6F2IFNH5hoGn4hcNdqGDzD3ZEI0Wr5WC3stUWbTj4HV7k7HCGEOKVVl1skp+WSV1RaeZKcnmx
WvJMk2bO5drgQQjQ5by9FsU1Dh2FwQEaShRDNR6seSa6+P/I681uSZCGEaDSBPl6cLLahOwyD43sh76i7Qx
JCCKCVJ8lr9x+nc2Qg0cF+p+9M2wABbSA0rsnjEkKI1iI2zJ/iUjvHIx0DEgdXujcgIYRwaNVJ8pZDOSS2DzVth35
+EbZ8Vbbz8AYziiyT9oQQotF0iPAHYJ93d7D6SsmFEKLZaLVJcmZeEWk5hfSNDYJv74HFT8G3f4CC41CcD0e3
SamFEEI0srhws5DPwVwbxA2BHd+D3ebmqIQQohUnyVvScrFg55IDz8L6D6H/NVCUC7++AelbQNskSRZCiE
YWF25GklOPF8CQmyFrD2yf6+aohBCiFXe3SE7N4kXvfxG9ZxmM+zOMexhKTsJvb4GymoPaDXBrjEII4ekCfL
yIDPLhYFY+jJ0C4Z3hl1eg90VS7iaEcKvWOZJsK2Xwuke43LoMxv/FJMgAYx+G4pPwy8sQGA0hse6NUwghW
oH24QEcPJ4PFiuMusd0F9q3zN1hCSFaudaXJNtK4ZvbGJb3I3Pa3AxjHyzbF5MAfS8DW7FM2hNCiCbSIdzflF
uAKX0LjDajyUII4UatL0ne+Als+ZLnSq7mUOKdp+8f+xAoC3QY0vSxCSFEK9QhIoC07AJsdg3efjD8DtjzExze6O
7QhBCtWOtLkg+spNivDW/bLjLt3yqK6gl3rIDhlSTQQgghGlxcuD8lNs2R3EKzYcjN4BsK8x8xLTqFEMINWl+Sn
LaewwG9AUWf2JDKj4nuDT4BTRqWEEK0Vh0cbeBSs/LNBr9QmPgM7P8FVv3LjZEJIVqz1pUkF+dDxjaS6Uxc
uD/hgT7ujkgIIVq9DhGOXsnOumSAAddCj4nw4+NwbJd7AhNCtGqtK0k+shm0nWUnO9A3tpJSCyGEEE0uNs
wPpTBt4JyUgov+Ad7+8M3tZtK1EEI0odaVJB/eAMCinPYkxkmSLIQQzYGvl5WYYL+yDhdOwW3hwpfg0Boz6
VoIIZpQ60qS09ZT7B/NUcLpW9mkPSGEEG7RIcLf9EquqO/lENMXfnsbtG76wIQQrVarS5LTAnoB0LeqSXtCC
CGaXFx4QNnEPVdKwbDb4GiyLDAihGhSrSdJLsqDjB1stnemU5sA2gT5ujsiIYQQDh3C/TmSW0hxaSUt3xKng
n+EGU0WQogm0nqS5CObAM1Pue0Z2CHM3dEIIYRwERcRgF3D4ZyC03d6+0PSjbBjHmTtbfrghBCtUutJktM
2APDLyTgGdgx3byxCCCHKiQv3Bzh98p7TkN+BxQqr3m3CqIQQrVkrSpLXU+DflgzCGCRJshBCNCvOBUUOVl
aXDBASCwlTYP2HcCK9CSMTQrRWrSpJ3u/TAz9vC73aBbs7GiGEEC7ahfphtajKO1w4nfUA2Evhk6lQdKLpgh
NCtEqtI0kuzIXMXawt7US/9mF4W1vHyxZCiJbCy2qhXWglvZJdxSTA1JlwZAt8fgPYSposPiFE69M6ssWM7QD
8nNOWgR3D3BuLEEKISnWODGRXel71B/U4Hya/AnsWwdw/Nk1gQohWqXUkyblpAByw tZFJe0II0UwN6B
DGjvQT5BfXsAT14Btg+J2w7r+QuadpghNCtDqtI0nOM5M80nUYg2QkWQghmqUBHcKw2TWbU3NqPnjk3
aCssHZmo8clhGidWkeSfOIwpXgREBpNdIifu6MRQohGpZSaqJTaoZTarZR6uJrjLldKaaVUkuP5P5uUqptUqpzY7
fE5ouapMkA2w4mF3zwSHtoNckWP8RlBQ2alxCiNaplSTJRzhGGAM7SamFEMKzKaWswBvABUACME0plV
DJccHAvcBKl83HgIu01onADcCHjR9xmTZBvnSMCGD9gezanZB0ExRkwdb/NWpcQojWqVUkyUXHD5FmD5
```

N6ZCFEazAU2K21TtFaFwOzgCmVHPck8DxwahhWa71ea53meJoM+CulfBs7YFcDO4ax/uDx2h3ceSxEdIU1
HzRuUEKIVqlVJMm2nMOk63B6S39kIYTnaw8cdHme6th2ilJqENBBaz23mutcDqzTWhdVtlMpdatSao1Sak1
GRsaZxnzKwA5hpOcWVb48dUUWi1mu+uBvkJ7cYDEIIQS0kiTZKz+ddB1OXFiAu0MRQgi3UkpZgJeB+6s5pg
9mlPm2qo7RWr+jtU7SWidFRUU1WHwDHN/41brkYsC1YPWFX14BrRssDiGEqFWSXNMkEKVUR6XUYqXU
eqXUJqXUhY7tbp0EAkBJAT4luRzV4cSENum3hkII4Q6HgA4uz+Mc25yCgb7AEqXUPmA4MMdl8l4c8A1wvd
a6yfurJbQLwcfLUrvJewABEabTxeYv4Ie/SqIshGgwXjUd4DIJ5FzM13arlVJztNZbXQ57FPhca/2WY4LIPCCesk
kgaUqpvsACKnzt1+hOHAHgpG8Uvl7WJr21EEK4wWqgu1KqMyY5vhq4xrlTa50DRDqfK6WWAA9ordcopcK
AucDDWuvlTRm0k4+XhT6xIaw/UMu6ZIAJj0JhDqz4J1i84ezHQKnGC1Il0SrUZiS5NpNANBDieBwKpEHzmAT
iTJJ1UNsmva0QQriD1roUuAszKLENM4CRrJR6Qil1cQ2n3wV0Ax5TSm1w/EQ3csinGdghnM2Hciix2Wt3glJ
wwQsweAb88jKsfq9R4xNCtA61SZJrnAQCPA5MV0qlYkaR767kOlVOAmmsCSAAnDgMgFdIu4a9rhBCNFNa
63la6x5a665a66cd2x7TWs+p5NhxWus1jsdPaa0DtdYDXH6ONnX8AzuGUVhiZ8eRE7U/yWKBSa+YjheLn4a
C7EaLTwjROjTUxL1pwEytdRxwIfChY3IIUPMkkMaaAAKgHUmyb5umrfIQQghRP85FRepUcgEmUT7vSSg4D
stfbfC4hBCtS22S5JomgQDcDHwOoLX+FfDDUfPm7kkgxcfTKNLetGnT5N8YCiGEqIe4cH/ah/mzeEc9vlls1x8S
r4Tf3oKcin9VCSFE7dUmST41CUQp5YOZBFLxK7sDwNkASqnemCQ5ozlMAinMOkS6DqOdtH8TQogWQSnF
pH7tWLozg+z84rpfYMKjoO2w5JmGD04I0WrUmCTXchLI/cAtSqmNwKfADK21phlMArHnHiadcNqF+TXlbY
UQQpyBi/rFUmrXLEg+UveTwzvBkFtgwyeQsbPhgxNCtAq1qkmuaRKI1nqr1nqU1rq/Y6LHQsd2t08CsZw8Qr
oOp32Yf1PeVgghxBno2z6ETm0C+Hbj4fpdYNS9ZjR56/8aNjAhRKvh8Svu+RVmkEE4kUGykIgQQrQUSikm92
vHij3HOJZX6crY1QuOgdiBsPuHhg9OCNEqeHaSXJSHr+0k+b7RWC3SWF4IIVqSi/rHYtfw/eZ6jiZ3OwdSV5tu
F0IIUUeenSTnpQNgD4xxcyBCCCHqqmdMMN2ig/h2U32T5HNNycWexQ0bmBCiVfDsJNnRI9kSIqvtCSFES6
OU4qJ+sazel8WRnMK6XyAuCfzCYPePDR6bEMLzeXSSbM91LCQSIQuJCCFES3TxgFi0hi/WHKz54IosVug6wS
TJWjd8cEIIj+bRSfLJY6kABLWJc3MkQggh6qNzZCBndY/ko5X7KbHZ636B7uea0rsjmxs+OCGER/PoJLkgK5V8
7Utkm4Zd6loIIUTTmTEynvTcIuZvqUfP5K5nm9/S5UIIUUcenSSX5hwmXYcRGy6r7QkhREs1vmc0ndoEMHP
FvrqfHBwDbfvB7kUNHpcQwrN5dJKsThzhKOHEymp7QgjRYlksiutHxLN2/3E2pWbX/QLdz4UDv8GJeoxECyF
aLY9Okn0KjnKMCEL9vd0dihBCiDMwNSmOQB9r/UaTB043k/h+/L8Gj0sI4bk8N0nWmsDiDPJ9o1BKFhIRQo
iWLMTPmysGx/HdxsOk59axHVxEFxhxJ2z8BA6ubpwAhRAex3OT5JIC/HQhtoBId0cihBCiAdw8ugt2rXlj8e66
n3zWAxDcDr7/E9jr0SVDCNHqeG6SXJAFgFdgGzcHIoQQoiF0bBPA1KQOfLrqAKnH8+t2sm8QnPsEpK2HDR8
3ToBCCI/isUlyaV4mAD4hkiQLIYSnuOfsbiil+OeieowmJ06FuKGw9IWGD0wI4XE8Nkk+mXMMAJ8gKbcQQg
hP0S7Un2uHdeTLdansPXaybicrBb0nQ/YByM9qnACFEB7DY5Pk/JyjAPgEy0iyEEJ4kt+P64aP1cKrP+6s+8kxfc
3v9C0NG5QQwuN4bJJclGvKLQLDZLU9IYTwJFHBvswYFc+cjWlsO5xbt5OdSfIRSZKFENXz2CS55IRJkoMkSRZ
CCI9z+5iuBPt68dKCHXU7MTgGAqMgPblxAhNCeAyPTZJt+Vnka19CQ4LdHYoQQogGFhrgzR3jurFo+1FW76
tjfXFMX0jf3DiBCSE8hscmyeRnkU0gEYE+7o5ECCFEI5gxMp7oYF+e/347WuvanxjTB45uB1tp4wUnhGjxPDZJ
thRlk0Mw/t5Wd4cihBCiEfj7WLn3nO6s2X+cxTuO1v7EtolgK4LMerSRE0K0Gh6bJHsXZXPSEixLUgshhAe7M
qkD8W0CeGnBztqPJsf0Mb+lw4UQohoemyT7luRQ4BXq7jCEEEI0Im+rhbsmdGfr4dzajyZH9gSLtyTJQohqe
WySHGDLpchbkmQhROujlJqolNqhlNqtlHq4muMuV0pppVSy7Y/O87boZQ6v2kiPjNTBsQSF+7Pa4t21240
2csHonpKGzghRLU8M0nWmkD7CUp9w9wdiRBCNCmllBV4A7gASACmKaUSKjkuGLgXWOmyLQG4GugDT
ATedFyvWfO2WrhjXFc2HMxm+e7M2p0U00fawAkhquWZSXJxHt6UYvcLc3ckQgjR1IYCu7XWKVrrYmAWM
KWS454EngcKXbZNAWZprYu01nuB3Y7rNXtXDI6jbYgf//xpV+1OiOkLJ9JkeWohRJU8MknWjg89S0CEmyMR
Qogm1x446PI81bHtFKXUICD1npuXc9trny9rNw2tgsr92axam8tEl/n5L0j0i9ZCFE5j0yS87IzALAESpIshBCul
FIW4GXg/jO8zq1KqTVKqTUZGRkNE9wZunplRyICfZi5Ym/NB7dNNL+l5EIIUQWPTJJPOpJkn6BIN0cihBBN7h
DQweV5nGObUzDQF1iilNlNoHDAfmOCbv1XTuKVrrd7TWSVrrpKioKIPEMv/78faxcMTiOhcnpZJwoqv7goGgIjI
aUxWC3N02AQogWxSOT5IIckyT7h7o0RodVYD3ZVSnZZ1cPHeOcO7XWXK11pNY6XmsdD6wALtVar3EcN1Up5auU6gx0B1Y1fRO+3dAlMsfPlbWSsuRGydn3Z
BC4BtwOda62Sl1BNKqYtrODcZ+BzYCswH7tJal505UOyGKjD/5QWm2wCvmnsXjPJPVP1JnTdJFl3dJbXwsgbY5rJVn82EG4X1
nsabrqTWSLdAbhgzPJNLTDARDfLrgADXwx85qYmOxx9LAlOCIBgNqONUUMJlOKyxB

SC8Y+Alx8s+j8IjYNzHm+yOIUQzZtHjiTb87PI036EhwS5OxQhhBBN7Pw+bQkL8OaTVQdqd8JZf4T+18CK1yF
zT+MGJ4RoMTwySbYUHiebIEL8PHKgXAghRDX8vK1cPiiOhclHOJZXwwQ+p3MeNyPK8//cqLEJIVoOz0ySi7LJ
U8EopdwdihBCCDeYNrQjpXbNv5fXoh0cQHAMjHsYdi2AnQsaNzghRIvgkUmyb3E2J60h7g5DCCGEm3SLDu
LCxHbMXL6PrJPFtTtp6K0Q2QPmPwwlhTUfL4TwaJ6ZJJfmUuQd6u4whBBCuNEfzu5OfomNd5el1O4ELx+44
HnISoEPL4WTmY0boBCiWfPIJDnQlkuRT5i7wxBCCOFG3WOCmdwvlv+s2EdmbWuTu06Ay9+HQ2vhvQmQ
8jMsexneGg3/uUgWHhGiFfG8JNluJ1jnYfeVkWQhhGjt7j27GwUlNt6p7WgyQOIVMGMuFJ+E/15s2sOVFsL
epbD928YLtiEUn4SiE+6OQgiP4HFJsi7KwYod7R/h7lCEEEK4WbfoYC7uH8t/V+wnt7Ck9id2GAK3LoGL/wn3b
oQ7V5p65cXP1rw6X/ZB2D4PdA2LmTSGL26E98+TFQSFaAAelyQX5poaMkuAJMlCCCHg2mGdKCixsWL3sb
qdGBoHg66H8HiwWE33i4xtkPxN5ceXFMLPL8DrQ2DWNPj+oaYtz8jPgt0/wtGtsPnLpruvEB7K45Lk3KyjAHg
FtXFzJEIIIZqDQR3DCPbz4uedGWd2oYRLIToBljxXfqRWa0ieDW8MhcVPQ4/zYMjvYNW/YM5dYCuBtA2w4p
9wcPXp17XVYYS7OjvmgbZBYDT8/FzDLLO9/1dZrlu0Wh632kZBjhkp8A2WJFkIIQR4WS2M7hbJzzsy0FrXv4e
+xQLj/gyfXwcLHoEOw8wCJMv/AQd/Mwn09f+DLuNM4hwYBUuehS1fQ2mBuYZvKNyyCCK7m+er34eFf4V
pn5jznDZ/CUW5kHRT7ePb9i2EdoQLnoNZ18DGT2HQdfV7rQAHVsK/J8KFL8HQW+p/HSFaKI8bSS7MNSMF
AWFRbo5ECCFEczG2RxRpOYXsPpp3ZhfqfZFJZle+DV/eaMoqslLgon/AbcvKEl2lTHnGlDeg31S47F245Sewes
On06AwBzZ9DnPvNwn0/+6GIkdsqWvg61th3oOQd7R2cRXmwp6fTHw9L4TYQab0o7SWPaIrs/lz83vL1/W/
hhAtmMclycV5piY5UJJkIYQQDmN6mL8TzrjkQim4bjY8uBfu+BVu+A7uWQeDZ4C1ki9nB043k//6XQntB8N
VH8Lxvaad3De3Q/xouO4byDkIi54wnSm+uhkC2oC9BNZ/WLu4di0EW7FJkpWC8X+BnAMw+3bYuRCK8+v2
Om0lpvba6gMHfoXcw3U7XwgP4HFJsv1kFgCh4ZIkCyGEMGLD/OkRE3TmSTKYJDQgAmISoPNZ4Btc+3M7jY
QLX4TDGyF2AEz71Iw+D73V1DB/fCVkH4Ar/wvxZ8GambXrVLFtDgTFmBIQgG5nQ9LNpsvGJ1Phhc6wcVbt4
0z5GfIzYfwjgDbXr46tBN4cAb+8Uvt7CNHMeVySrPOzyNUBhAb6uzsUIYQQzcjYHlGsTMkiv9jNE9GSboIbvjUj
0s4E++zHIKwjHFgBY/4EnUbAkJvNaPDuH6u/XnE+7PoBek02ddNgEvnJL8ND+8xIdfskmP172PZd7WLc/AX4
hcHwO02tdVUdPZx2/WC6aix+1pSfOGXugR8eg9y02t1XiGbE45Jka+FxclQwXlaPe2lCCCHOwJgeURTb7KxM
yXJ3KNB5DPiFlD33DTKjx2fdD2MeNNt6TTajw2s+qPo6uYdhxWtQkm9KLSry9jOrCF7zGbQfZOqoU5ZUH1tx
Pmz/DhKmmKW6Ey6BA79VX3Kx8RPwjzA11/MfMdtOZsJHl5mJjW8Mh/UfnXnv6CObyyfhQjQij8skrcU5nL
TU4asvIYQQrcKQ+Aj8vC0NU3LRGGIHmhFlZ22z1dv0ad65wJRgOGltul+8MQxe7mU6aMQkmvrmqvgGwT
WfQ5tu8Ok1cHhT1cfunA/FeWblQYA+l1Cu5CJlCWz4tOz4/CzYMR/6T4OxD8LO702njc+mm8T68vehbV/43
53w8VQ4Wcd+1a73mTkJPrvOPQu1iFbH41rA+ZbkcMIaUvOBQgghWhU/bysjurRhyY6jaJ1Q/1ZwTWnwDFj
2d5MY9rsS2g2ApS9CymJo2w/OfRK6jDVJsqWGca+ACFN68e4E0yLulsUQFGUWPFn1L9PLObitWX47uB10G
mXOi+ppSi7WfWhKP3YtNNv9QqDXJJOw20tgwDVmVcL1H8HnN5iezZe/b5LtPpfB6ndNu7u3z4KpM6HjsLq
9Fz8/b7qCFObAvl9MPbgQjcjjRpK9bQWUege6OwwhhBDN0IRe0ezLzGdPxkl3h1I7oXEw6e9QWmh6M8+8
0LSIu+BFs2z2qHugXf+aE2Sn4LZw9cdwMgM+vx5OpMOnV8H8h03i/esbkLbOJLwWa9l5CZdA+mbTO/ncJ0
2CPuceyMuADR9D20QzWuzlY/oqA4x9uGw02mKBYbfBzQvNCPnMC2HWtWU/B1ZWH/ex3bD6PTNa7R9h
WvAJ0cg8byTZXkCpRSbtCSGEON3ZvWP46/+SWbQtnW7RQe4Op3aSbjI/x/fBwVWm60VIu/pfL3ag6d/81c
3wj36me8aFL5lVAgEKs82kPVfDbwe/UJP0BkZC9/PgX2Pgkyvh8AaY+FzZsV3Gwp92m5Hr0+49AG5bCt8/aE
auLVbIS4eDK+GOFRAUbY6zlZpWeRFdzDE/PGYWbjn3CQiJhWUvm/cjPL7+74MQNfC4kWQfirBZ/dwdhhBC
iGYoNsyfPrEh/Lgt3d2h1F14vCm5OJME2SnxCtNLOTweblpgVtRTyvz4h5vfrvxCTaIcGGmeR/eCc/5mRp0tXp
A4tfzxlSXITv5hcNk7cNcq+P2vMGOu6Q89+w5T+lGYYyb8vZ4EL3YzI8075sJZfzRJ9JDfgbLAqnfP/H0Qohoely
T76SJsXpIkCyGEqNw5vWNYu/84mXlF7g7FvcY+CHeuhLjB9Tt/2B1mdb8B15Qlz/UR3RvOf9rUOy/6P3j/fNi/
HMY+BN3OMfXH4Z1h+O/N8SGxpvPGug/LVikUohF4VrmF1vhRhM0rwN2RCCGEaKbO6R3DPxbtYvGODK4
YHOfucFoui8UshtIQkm6G3Ytg+avgGwrTvypb4ttWCtpu6p2dht8ByV/Dwr+YUhGrt9lefNJ01Ijs1jBxiVbNs0a
SS4uwoLFLuYUQQogq9G0fQkyILz9ubYElF55KKVMnPfRWuHlBWYIMpiWea4IMEDfELHSydqZZ4jv7oJnY99
pAeGOoqVcW4gx5VpJcYtam114ycU8IIUTllFKc0zuGpbsyKCypxZLPomkERJglu6N713ysUjDxGdNi7vBGeDU
R5t5vVi3UNrNioBBnqFZJslJqolJqh1Jqt1Lq4Ur2d1RKLVZKrVdKbVJKXeiy78+O83Yopc5vyOAr0sWmpY/2lnIL
IYQQVTsnIYb8Yhu/pmS6OxRxJhKvgFt+MhMHp30GN/8AHUfCps9lwRFxxmpMkpVSVuAN4AIgAZimlEqN

ijwOda64HA1cCbjnMTHM/7ABOBNx3XaxSlRY6RZG8ZSRZCCFG1EV3aEOBjlZILTxDdGy5/F3pONCPM/abCs
Z1mhLm2Dq01fZ/t8s2CKFObkeShwG6tdYrWuhiYBUypcIwGnMvchQJpjsdTgFla6yKt9V5gt+N6jaK00DHLV
UaShRBCVMPP28rYHlEs3JqO3S4jjh4l4RKweNet5GLZy7DuP3B0a6OFJVqe2iTJ7YGDLs9THdtcPQ5MV0qlAv
OAu+twboMpKXSsoCRJshBCiBpM7NuWjBNFrDtw3N2hiIYUEAHdz3Usl12LkeGCbNj1g3m8/9dGDU20LA01
cW8aMFNrHQdcCHyolKr1tZVStyql1iil1mRkZNQ7CJuj3EL5yLLUQojWqxbzSG5XSm1WSm1QSv3iLKFTSnkrp
f7j2LdNKfXnpo++6UzoFY2P1cL8LUfcHYpoaP2uhLwjsG9Zzcdu/w5sRWD1hQOSJIsytUlkDwEdXJ7HOba5uh
n4HEBr/SvgB0TW8ly01u9orZO01klRUVG1j74CZ7mF8pGaZCFE61TLeSSfaK0TtdYDgBeAlx3bpwK+WutEYD
Bwm1IqvkkCd4NgP29Gd4/k+y1H0DLJy7P0mAg+wWYCX002f2kWK+l9kUmS5b8F4VCbJHk10F0p1Vkp5YO
ZiDenwjEHgLMBlFK9MUlyhuO4q5VSvkqpzkB3YFVDBV+RzdHdwuIrI8lCiFarxnkkWutcl6eBmHklOH4HKqW
8AH+gGHA91uNM7NOWQ9kFbDnk0S+z9fH2NxP4NnxieilXJe8o7P0Z+l4OnUbAicPSY1mcUmOSrLUuBe4C
FgDbMF0skpVSTyilLnYcdj9wi1JqI/ApMEMbyZgR5q3AfOBOrXWjTR21FzmSZCm3EEK0XrWaC6KUulMptQc
zknyPY/OXwEngMGbw4yWtdVZlN2moMjl3OzchBqtFMT/5sLtDEQ3t/GfMstbf3gu/vVX5McnfmNX8EqdCx
xFmW1UlF0V5MsrcytSqblhrPU9r3UNr3VVr/bRj22Na6zmOx1u11qO01v211gO01gtdzn3acV5PrfX3jfMyD
HuxqUn28pOJe0IIUR2t9Rta667AQ5g2nmBGoW1ALNAZuF8p1aWK8xukTM7dwgN9GN4lQkouPJG3P1z9
MfSaDPMfhvmPQNGJ8sds/hJi+kJ0L4jqDX5hlSfJqWvgxa6w/qMmCV00Dx614p52JskykiyEaL1qNRfExSzgEsf
ja4D5WusSrfVRYDmQ1BhBNicT+7YjJeMku4/muTsU0dC8fGHqfyDpJvjtDfhnkkl0k2ebpDl1lSm1ALBYoOPw
0ztc5B2Fz66D0kLYMa/JX4JwH49Lkku1BW8fX3eHIoQQ7lLjPBKlVHeXp5OAXY7HB4AJjmMCgeHA9kaP2M3
OT4hBKZizMa3mg0XLY/WCya/A7xZBaHv4353wxQ2w+j3oMg4GXV92bMfhkLkL8hwlRLYS+GIGFByH+LNg7
zKwlbrjVQg38HJ3AA2qpIB8fPHxbrRF/YQQolnTWpcqpZzzSKzAB855JMAaR5ncXUqpc4AS4Dhwg+P0N4B/K
6WSAQX8W2u9qelfRdOKDvFjfM9oZq0+yN0TuuPj5VHjR8IpLglu/hH2LgHfUGibCF4+5Y/pONL8PvibqWf+/k
HYvxwufceMSn9xg1mdr+OwJg9fND0PS5LzKcRXPuCEEK2a1noeZmEn122PuTy+t4rz8jBt4Fqd64Z34saZq1
mQfISL+se6OxzRWCwW6Dqh6v2xA8DLD1b+y5Rj5ByAUX+A/ldBfhagIGWxJMmthEdlk6oknwLtg4/Vo16W
EEKlRja2RxQdIwL48Lf97g5FuJOXL7QfbBYh8Q2GGXPh3P8z+wIiTBK9Z7FbQxRNx6OySeUst5CRZCGEEHVgs
SimD+/Iqr1ZbD8iPZNbtfOfgUv/BbcthfjR5fd1GQ+pq6FQ/htpDTwqm1S2Aim3EEIIUS9TB3fA18vCh7/KaHKr
FjsA+l9tJvxV1HU8aJupUxYez6OySWtpAQXaB18vmbgnhBCibsIDfbiofyzfrD9EbmGJu8MRzVGHYeDlLyUXrY
THJcn5+OIrI8lCCCHqYcbIePKLbby3NMXdoYjmyMsXOo2ElCXujkQ0AY/KJq3OcguZuCeEEKIe+rYP5eL+sfxra
Qqpx/PdHY5ojrqOh2M7IPtgzceKFs2jskkvWyGF+GKxKHeHIoQQooV6+IJeKAXPfe/x66iI+ug1GSxe8MvLDX
O9b26Hz6Y3zLVEg/KsJNleSJHFz91hCCGEaMFiw/y5bUxXvtt0mNX7stwdjmhuIjpD0s2wdiYc3XZm10rfChs/
hW3fwYn0BglPNByPSpK9bQWUKFmSWgghxJm5fWxX2oX68X/fJlNcand3OKK5Gfew6aO88K9ndp1fXgarD
6Bh+7cNEppoOJ6TJNvteOtiSmQkWQghxBny97Hy18kJbDmUyx8/34DNrt0dkmhOAiJgzJ9g9w+w6wfYOgd
mToYPL3OszFcLWSmw5SsYeitE9oCt/2uY2LL2gpb/XhuC5yTJpQUAFEuSLIQQogFcmNiORy7sxXebDvPo7C1
oSTyEq6G3Qng8fHwFfH4dHN9nVur7YCJkH6j5/F9eBYs3jLwbEqbAvl8gL6P+8djtsOAv8NoASP66/tcRp3hOk
lxsZiHbrJIkCyGEaBi3junKXeO78emqA7y4YIe7wxHNiZcvXPSamch31cdw70aY/jWWcOALvnQs/vwiLnoAf/mZ
GjV3lpsGGT2DgdAhua5JkbYft39UvlpIC+HIG/Po6oKRFXQOpZDmZFqrEJMkl1gA3ByKEEMKT3H9eDzJPFvP
mkj2M7BrJ6O6R7g5JNBddxpofp85nwU3z4dOrYPFToCyAMgnxdd9A274mif78epMUj7rHnBfTFyK6mpKLp
BvrFoPdBh9PNSPR5z0Ne5fC/l8b7CW2Zp4zklxiyi1kJFkIIURDUkrxt4sS6BIVyINfbpTV+ET1YhLgng3w6FF4LA
t+/5tpGTfzQljzAfxrrOlqMfXfplwDQCkzmrx3KZzMrNv9Nnxsyjwu+geMvAs6jYDMXXDyWNkxKT/D4mca6h
W2Gh6UJJ8EwG71d3MgQgghPI2ft5W/T+3PkdxCnvpuq7vDEc2dxWrKMZSCqB5mdDmgDXx3H3j7we9+NE
mxq4QpoG3w6z+h6ETt7lOYa0o6OgyDQdebbR1HmN8Hfis7bslz8PPzkHv4zF9bK+JBSbJjJNlLkmQhhBANb2
DHcG4f25XP16/Ai93gs+tq7p287CU4mQETnzUDsB5GyBAL1u3OUDsQ
LD6wgFHyUXu4bLHe39usJfUGnhOkuyYuKe9JUkWQgjROO49pzu924Vw76wNbErNdnc4oiUJjoHR95n2cZV
RCmbMNcn0oBtg53wz+luVrBT47S3ofw20H1y23csX2g8qG0neNgfQ4OUPexY32MtpDTwnSXZM3JNyCyGE
EI3F18vKv2cMISzAmxs+WMWu9Fp+LS5EbVgspqb4whcg8UqzGl9Bdtn+9R+bUebnOsGbI0wLubMfO/06H
UfA4Q1mADH5G4hOgJ4XmK4X0sqw1jwoSTblFtpbulsIIYRoPG1D/fjo5mF4WS1Mf38lKRl57g5JeKJht5oBw

PUfmed5R+H7hyAkFvpdBYNvhGmfQEi708/tOALspbB9rhlR7nMpdB0PeUfOfCntVsSDkmQzcU9JuYUQQoh
GFh8ZyEc3D6O41M6UN5bzw1apURYNzFmjvOod0+Zt0RNQWghX/NuMNF/wHHQZV/m5HYYAChb9H6Ah
4ZKyY6WHcq15UJJsRpLxkZFkIYQQja9n22Dm3DWa+DaB3PLfNby4YLssXy0a1rDbIHu/qU1e/xEMvx3adK35
PP9wU2KRc9D8juoBYR1NL+YUqUuuLQ9MkmUkWQghRNPoEBHAF7eP4OohHXhj8R6mv7eSoycK3R2W8B
S9JkNIe5MkB7SBMX+q/bkdh5vffS4t29Z1POxbDqXFDRunh/KcJLn4JMXaC29vH3dHIoQQohXx87by3OX9e
PGKfqw/eJwL//ELK/Ycq/lEIWpi9YIhvzOPz34M/EJrf263c8DqA30vL9vWZbwpT01dVf7YohOQuUcm9VXgM
ctS65J8CvDB1+o5eb8QQoiWY2pSB/rFhXHHx2u59r2VXD2kIw+e35PwQBm8EWdg+O+hTTczqlwXPS+AB3a
Bf1jZts5nmaWyl/8DtnwNGTvM6nx5jpr6SS/DkJsbLPSWzmMySntxAfn44ePlMS9JCCFEC9OzbTDf3jWa343
uzOdrDjLh70uYszHN3WGJlszbDxIuNu3h6kKp8gkymJHoTqNg10LY/CXYS6DbuXD238yqfYuegPysBgu9nOJ8
eO9cSJ7dONdvBB4zkmwvOkmB9pEkWQghhFsF+nrxl0kJXD44jke+3sy9s9ZjUTC5X6y7QxMCps0y5RXBbct
W6QMz8vzWKPjpSZj8SsPfd80HpsxjeSn0uaThr98IPCaj1CX5FOKLj5RbCCGEaAZ6tQ3hk1uGM6RTBH/8bCM
rdkudsmgGfINMb2XXBBkgujcMvRXW/BsOb2zYexbnw/JXwTsA0tZBenLDXr+ReExGqYtOko8vvt5Wd4cihB
BCAGZS37vXJ9E5MpBbP1zLzOV7+Xz1QeZsTCOvqNTd4QlR3riHTReN7/4Ih9aBrYH+G13zPpzMgMvfM6sEr
v+4Ya7rdHhjoyTeHpMkU1Jgyi1kJFkI0coppSYqpXYopXYrpR6uZP/tSqnNSqkNSqlflFIJLvv6KaV+VUolO47xa9
roPU9ogDf/uWkoEYE+PP7tVh78ahP3fLqeK95aweGcAneHJ0QZ/zCY+CwcWgvvjofnOsCn0yBtQ/2vWXzSTB
TsMh56TYJeF8KmWZW3obOVQGFu3a6vNcx9AD6+suGSegfPyShL8inAV2qShRCtmlLKCrwBXAAkANNck2C
HT7TWiVrrAcALwMuOc72Aj4DbtdZ9gHFASROF7tHahvrx4x/HsvKRs/nlofG8c91gUo8XcNmbK9iZfsLd4QlR
pt+V8MetZmW/gdPhwK/wzlj4YgYc31/36612jCKP+7N5PvA6yM+EnfPLH5e5B94aCW+PhtKi2l9/1w+m1nn
M/aZlXgPynIxSkmQhhAAYCuzWWqdorYuBWcAU1q5DNYGAsznqecAmrfVGx9TcKvh4W
YgJ8SMuPIDz+rTls9uGY7NrLntzBXd+so4PftnLjiOSMItmICQW+l4GF74I9240i5jsXAgfX2FGe2vDbodlL8OPj5
uezR2Hme1dJ0BwrFlB0GnPYnh3AuQcMisMbvy0bJ/WsOP7yrtu2O1momF4vEm+G5jHdLdQpabcIkSSZCFE
69YeOOjyPBUYVvEgpdSdwB8BH2CCY3MPQCulFgRBwCyt9QuV3UQpdStwK0DHjh0bLPjWpE9sKF//fiQvLdj
B6n3HmbvpMACT+7Xj4Qt6ERce4OYIhcC0jZvvwKMQOhFnXwNqZMPQWs+/gKvhsOmg7BMWYjhmRPc0y2
Fv/B3t+Miv+XfSPsutZrDBgGvzyCnx0OeQdNfXEkT1g2qfw5Y3wy6swYLoZGd7yFXx1M/ScBNM+KR/b9m/hy
Ca45G2wejf4S/eYJNlSWkABvrKYiBBC1ILW+g3gDaXUNcCjwA2YvxNGA0OBnFcRUmqt1npRJee/EbWtcuSWR
9km2c056Fj2ugC8/GDyqzB4xumdNAbPgJ0LzOhwcDvoPAbGPgR+IXDW/SbxTv4G4kfD3PvBJxh2zIWUJdBln
LmG3QaLnzHJdb8rG+Ule1ySLOUWQohW7hDQweV5nGNbVWYBbzkepwJLtdbHAJRS84BEoEIgrrWeCcwESEtLk
+eO5Pbh6SAdeXLCD//66jw+W76VX22C6xwSjtcbLorh8cBxndY9yd7iitVEKzn8K3hkHS1+Co1tNrfHNC80os5
PdDjkHwMsfgmMqv1ZYR7hjeeX7ek6CqN6w7O+w+XNTn3zLIvj0apj/CNy21IxGL3sZMrbD1JnmeSPwjCTZV
oLFXiKLiQghBKwGuiulOmOS6nOAa1wrKKW6a613OTYvBJyPFwAPKqUCgGJgLPDvRolaANDqaIBfnnsq7t3+Ztt5zbe2AlTJn+0G6
/Rb/0+j+rduI5f6UEKAwx3U/qZUT9hqW1aqqpE0qi7ZKCq9cnOPKN8+eVqwbw1
8kJzN2UxpyNaWxOzUYpRW5BCbM3pDEpsR1/mdQbq0WReryA3MISQvy8CPHzpn24PwE+nvHXu2hmYgd
Cv6thxWvm+aS/l0+QwawMGB5f/3tYLHDWH+HrWyBjG0x8HmL6wLlPwhc3wOr34Mhm2PARJFwCvafUe
Mn68oz/i0pMC518fPH1kj7JQojWS2tdqpS6C5PwWoEPtNbvKngjDcX5Ybrty7VSf7x31J7k0w+pB+kMgXv8q+ZUW/n5+Y2PGX+m3QVNK2STYmVW/jg9wqeSttuWiNSmXHAsZf
8bzXnfoRpAyeaIT9vK3ef3Z1LBrbnm/WHCA8wI8eh/j7kFZWSnV/Mom1H+WD5Pj5Yvo/4NgGEBfgQHuDD1
KQ4zkuIQVWsDxWiLkLbwx82gV/Y6bXGDcXqBbf8ZOqdnUmwUnDhC/D1rabbRt/LGufeLjwqSZZyCyGEEK1
Bh4gA7jm7e6X7pgxoz8MX9OKTlQfYe+wk2QXFbDucy20frmVofAR3TuhGTkEJ2w7nkplXRHSwHzEhvoQF+
BDgY8Xfx0r/uDACfT0jRRCNwD+88e/hXcm3IG0T4fe/Nv69HTzj/4BikyTn4ycr7gkhhGj1YsP8eeD8nqeel9rsf
LbmIK/8sJMbPlgFgJdFER7oQ9bJz28g1K2oX68cSUvpybUDbxym7XWCwyCi1aD89Ikh01yYXISLIQQghRk
ZfVwrXDOjFlQHtW7D5Gh4gAukYF4eNlwWbXZOYVkV1QQkGxjaMnivj7wh3c8t81nN0rGm+rhU2p2WQXlP
DnC3szfVhHKdkQrYKHJMn/396dx1Vd5Y8ffx32VWRzBQV3JbyyuC9Wk1De9PUfJvq1
6Qz5mRqRWPYaOJomeSu5YpL5UYuYOJCiIoiKss9vz/u9QYIagpe78f38/HgEZ/zD7nwnQL

d9wlSRZCCCFq5OPuwu0RTaqUOTspGjXwoFEDD1tZfPtgPlyznynL9xLk605MWADHz5znpfnbWb+vkCcT2rBi
dwFf/XiY8gpN37ZB9G0bTGRzP/y93a53tYSoF4ZKkktwl+kWQgghxDVydXbisfg2jItrbRs1Nps109bs5+3MbN
uTNaJaNKSBhzOfrD3Ah2tyAPDzdCU8yJu4dsEkdm5K28a+dquHENfCUElymXKX+VJCCCFEHak8rcLJSfFoXGt
6tgpk688nSOjYmNAAy9LZZ0sr2JR7nJ/yT5Nz7Ay7j55m8vI9TFq2h3aNfUju0pwkUzPb8UI4AoMkyZY5yeUu
8jxIIYQQoj6ZQhtiCm1YpczTzZl+7YLp1+7XlQB/OXWOxTuOsvD7w7ydmc3bmdl0aOJLy0AvWgR40adtMP3a
Bsn8ZnHDMkaSbH26hdnZ4zIHCiGEEOJ6aNTAgwd7hvFgzzDyTpSw4PvDbMo5zr6CM6zMLuDDNTn0aRPEc
3d1oFPTBpw6V865sgoa+bpL4ixuCMZIkq3TLSpkJFkIIYS44YT4e/FYfBuIt2yXlpuZteEAk5btYdDkb3F2UrbH0E
U0a8ADPSxP4vB0k1V0hf0YKknWLjKSLIQQQtzo3FycGN07nCHRIfxn/QFKSsvx93JDa/hiSx6p837ktS93EtWiI
TEt/OkWHkhsmD8erpI0i+vHMElyqXLH1cUY1RFCCCFuBn6erjzev02Vsof6hpN14AQLvz/M5gMn+NeKvZiX7
8XD1YkerQJpFeSDUpbFUG6PaExMywA7RS+MzhhZZdlZzit33FzkX5hCCCGEI1NK0TUsgK5hluS3+Hw5m3KO
s+qnAlbvKSAr9wRaa0orzHywej933tKEPw/sQFiQt50jF0ZjjCQ5PI5vfiqT1faEEEIIg/Fxd6F/h0b079CoSnlJaTkf
rs7hg9X7WLIzn7simzKyV0uiW/jLjX+iThgjSe6UxJxvg3E32zsQIYQQQlwPXm4u/HFAW1K6h/LBqv38N+sgC7
4/zC3NG/BYfBsGRjSRtRPENTHM0GtpuRl3V8NURwghhBBXoJGvBy8ldmLD8wn8bXAkJecreGzWFm5/bzWz
N/7MyZJSe4coHJRhssrSCrMsSS2EEELcpLzcXLivewuWPBXHv+6LwsVJkTrvR2JeX8oDMzYwf+shyiosXzmbzZ
r/bTvEmI838VP+aTtHLm5UxphugWUkWeYkCyGEEDc3ZydFYudmDIpsyvZDp1i0/QiLfjzChM+38XZmNvd1b
0HmjqP8kFeEk4Kdh08xd1xPQvxlyWxR1RVllUqpgUqpbKXUXqVUag37/6GU2mb9+UkpdbLSvreUUjuUUru
UUpNVPc2mlyRZCCGEEBcopYgM8ePPAzuw4ul4pj8YS1M/D97OzObY6fP8v+EmFj7ZhzOl5Tw4YyOFxec5U
HiGGd/m8MXmPHuHL24Alx1JVko5A1OA24A8YJNSaoHWeueFY7TWf29yso/S0QB6yso/
htSstluoUQQgghLubkpBjQqTEDOjUm99gZmvh52BYmmTmqKyOmb6DfWys4U1phO8fFWZHcpbm9QhY3g
CvJKrsBe7XW+7XWpcBsIPZqAUf/Xh1q60QkaShRBCCHFpYUHeVVbu6xoWwAcPxN
CzdSCv/K4Ty5+Oo1t4AM/O/YHvD560X6DC7q5kTnJz4GCl7Tyge00HKqVaAuHAcgCt9Tql1ArgCKCAf2mtd9
Vw3sPAwwAtWrT4LfHbnC+TJFkIIYQQv118+0bEt//1OcxT748mecp3jE3L4o3BkZw6W8apc2XceUtTmvh52
DFScT3V9Y179wJztdYVAEqpNkBHIMS6f4lSqq/Wek3k7TW04BpALGxsfpqXvi8jCSLelJWVkZeXh7nzp2zdyji
BuLh4UFISAiurq72DkUIIUccCfdyZPjKWoe+vZWxalq3836v2MWNkV25p7mfH6MT1ciVJ8iEgtNJ2iLWsJvcCj1
faHgys11oXAyilvgZ6AmtqOPeqaa0tz0mWZalFPcjLy8PX15ewsDCUUnYPkUrQRR0Cm9pEgvvWLTrevzLJ5mfs5ph+Y3rzLkXsSjJaEQUHTwZNamhuqZSwqXcnH8CC1qZPd3yu4yzgHCmVicUh5RbjuR7iNI2c1nxfaelXqz4qu8QEvgIpPl/B5r6A6Lh9J+lSrqqSvjRcS68eOxTLOJ3XmfqaOZmaDrY+9aZUZnNZqPvNaHiqU4Y7JzZXJS8IWBGjwZHuSawd/ScYHsvoHU9HrrIYoV43pcyQvYI2zYGoq0+NKOmehx+4d+rZ6xOQULdULuZqk9oXUVxQt2v5M8cRs84ejo5CddlK8IXSeeQbU8ZQ2ITLmEPIhkVdGqjMzEpUBLIhh6CwkZjSG8IbiYyU8QIQwn7zf+cQ8c0E7RAJ8+6JKZrVT1vnb8CY5nHmh3xtKQ5oa1zaRbzjMWLUsIl+TZeSnt5EoZ7YKO1QEZ7ZgRGjY3iInnBmT1ZgMKGfFhz8JPcaPb4M9PA+znfuUKnXR6+UPW9tr1+ohUiiPmqY5g0ZRqlq4Y3PSxQ

UREjR45kz549KKUoK6v5DutBgwbh7u6Ou7s7jRo1Ij8/n5CQkCrHdOvWzVbWpUsXcnNz8fHxoVWrVrbENC
Ulpcqo7QWlpaUsWrSId999F19fX7p3705mZiaJiYksX76ctLQ0AJydnfHz8yMtLY3hw4cTFBQEQEBAwGXr3a1
btyqPXZs8eTIZGRkAHDx4kD179lBQUEC/fv1sx1247pgxY0hOTmbChAnMnDmT0aNHX/b1HI1SyhmYAtyGZf
GnTUqpBZWTYOAzrfW/rccnAe8CAyvtfxf4+jqFLISShOTsperYOpGfrQJ64tQ0vZGzn9a928cYiy8O+fte5GcNiQ
gjyccfJCV7I2M4Tn21l84ETPH17e3zcDZGqOSRDvPMXnm4hj4AT9e1qRnzri7e3t+33l156if79+5ORkUFubi7x
8fE1nuPu7m773dnZmfLy8qs6pjaZmZmcPHmSyMhIAEpKSvD09Kx1akZtXFxcMJstf9dms5nS0lLbvsr1XrlyJU
uXLmXdunV4eXkRHx9/yceyhYaG0rhxY5YvX87GjRuZNWvWb4rLQXQD9mqt9wMopWYDyVhGhGHQWlf+O
sQby9OHsB5/N5ADyHe+QtSxpn6ezBgZy8IfjrDt55Pc1z2UNo18qxwz++Ee/N+i3cz8LofPNvzMrR0akdylGbd
3aoKTk+UmwLIKM/9ctoejp87x+9hQYlr6yw2C9cAQWaXcuCdudkVFRTRv3hyAjz/+uM6v3759e/bv309bi4
An3/+eY3HpaenM336dHJzc8nNzSUnJ4clS5ZQUlJCQkICU6dOBaCiooKioiJuvfVW5syZQ2FhIYBtukVYWBibN
28GYMGCBbWOjBcVFeHv74+Xlxe7d+9m/fr1APTo0YPVq1eTk5NT5boADz30ECNGjKgyEm8wzYGDlbzrGV
VKKUeV0rtA94CxlvLfIA/A3+5DnEKcVNSSpFkasbLv+t0UYIM4OrsxMu/68S8x3pxb9dQNuWe4NH/bOHeD9
ezv6CYwuLzjJi+gcnL9/LlD0cY9u913PHeajK25mE2X9WixaIWhsgqLyTJMpIsblbPPvsszz33HFFRRUb9p5PdKeX
p68v777zNw4EBiYmLw9fXFz6/qswqwlJSUsXryYQYMG2cq8vb3p1asXCxcu5JNKkSaxYyLEyhYyLYuXMnERE
RvPDCC8TFxWEymXjqqacAGDt2LKtWrcJkMrFu3boqo8YVDRw4kPLycjp27Mi4bFoqo8evQAIDg4mGnTpjFkyB
BMJhP33HOP7ZykpCSKi4ri4sNOdXit9BaT9Fat8aSFL9oLX4V+MeFVVIvRSn1sFIqSymVJTc/ClH3olv485fkW9jwf
AJvDe3M7iOnGDhpDXdOWsO2gyd5754uzZL04gL8PjcTZyYmJn3/PkKlr2frzCXuHbhiq6tof9hcbG6uzsrIuf2Al
S3fm81BaFguf6ENkiKynLurWrl276Nixo73DsLvi4mJ8fHzQWvP444/Ttm1b/vCHP9g7rCuWlZXFH/7wBzIzM2y8kSlUpdMYxTcHp88S29Ce5SzPuimxKoI977RcVl+yzDTUnWaZbCFF/P
vzwQz755BNKS0uJiorikUfqfZ2gOvfmm28ydepUo85FvmAT0FYpFQ4cwrIa6n2VD1BKtdVa77AGit+1
Y65lWg+HIJshDi+mjk68GU+6Nr3OfkpBgeG8qdkU35dN0BMrbm8dL/dvD6V7v44IEY4ts3us7RGoMxkmSZk
yxEvZs4caJDjhxXlpqaSmpqqr3DqFda63Kl1BNAzN0ZHwM3UWu9QSr0GZGmtFwBPKKUGAGXACWCk/SIWQt
QVH3cXxsW3Zlx8a3YdOcXT//2eRz7/87x7jc/kXeyhNbBPrRp5MPtnZrQxM/D3qHfkAyRJM
tzkoUQ4lda60XAomlL1f6/Y9XcI1X6z4yIcT10rFpAz79b8QirFpAz79QzfunbaeP3yyiRE9WjJr/QHKKjTtm/gyb8shis+X87d
Fu3g0rjWP9GuNp5shb2a+aoZIkmXFPSGEEEKIqgJ93Jn1UHd+/8E6pq3ez4COjXzRWrP/2Bne/eY
n3lu6h883HWRM73CGxYTg7+2G1pqcY2dwUoqwoJpvnjY6QyTJFxYTcXeVJFkIIYQQ4oJGDTz4Ylwlwvcg
tLiGnpbytXStE62Icp90czMuc4by3ezRuLdvH2N9l0DfMn++hpjhWXohT8oXc4T9/e/qYbaTZEkmy7cU9GkoUQQggh
qgj0cb/kUy66hQcwd1wvpvdvh05xWcbfmZjznH6tQ2ma3gA2w8VMf3bHJbsyie5S3NOnCml8Mx5is9XcK6sgrIK
M4183Qn196JZQ08aernSwMOV8GBvWgf7XMda1j1jJMky3UIYWP/+/UlNTeWOO+6wlb333ntkZ2fbFueoLj
4+nnfeeYfY2FGFjuusPvvsMxo2bFjlmPHjx+Pj48PTTz9dq3vPnTuXdu3aERERAcBLL71EXFwcAwYMqINPKH/94bz3b8r+/vXlTn5Ok/rzekXXpBfE6rXkI0APWcnMPqVaGwlG/Ldmr8p+6+Xn1mvEVH7
Y65lWg+HIJshDi+mjk68GU+6Nr3OfkpBgeG8qdkU35dN0BMrbm8dL/dvD6V7v44IEY4ts3us7RGoMxkmSZk
yxEvZs4caJDjhxXlpqaSmpqqr3DqFda63Kl1BNAJpZHwM3UWu9QSr0GZGmtFwBPKKUGAGXACWCk/SIWQt
QVH3cXxsW3Zlx8a3YdOcXT//2eRz7dzMeju9GzdSD5p87x7jc/kXeyhNbBPrRp5MPtnZrQxM/D3qHfkAyRJM
tzkoUQ4lda60XAomplL1f6/Y9XcI1X6z4yIcT10rFpAz79QzfunbaeP3yyiRE9WjJr/QHKKjTtm/gyb8shis+X87d
Fu3g0rjWP9GuNp5shb2a+aoZIkmXFPSGEEEKIqgJ93Jn1UHd+/8E6pq3ez4COjXgpsRMtA73RWrP/2Bne/eY
n3lu6h883HWRM73CGxYTg7+2G1pqcY2dwUoqwoJpvnjY6QyTJFxYTcXeVJFkIIYQQ4oJGDTz4YlwvcgtLiGnp
bytXStE62Icp90czMuc4by3ezRuLdvH2N9l0DfMn++hpjhWXohT8oXc4T9/e/qYbaTZEkmy7cU9GkoUQQggh
qgj0cb/kUy66hQcwd1wvpvdvh05xWcbfmZjznH6tQ2ma3gA2w8VMf3bHJbsyie5S3NOnCml8Mx5is9XcK6sgrIK
M4183Qn196JZQ08aernSwMOV8GBvWgf7XMda1j1jJMky3UIYWP/+/UlNTeWOO+6wlb333ntkZ2fbFueoLj
4+nnfeeYfY2FjuuusuPvvsMxo2bFjlmFdffRUfHx+eeeaZWl97/vz5tGvXjBLdJQnoE6o6FOzCn2uwOGaPGWQzFe
mTJjAnDlzOHjwIE5O8vQiAwXhEnvjrzY6S/Ks8VHnPbjywebUt/Yb/fz9T9UtLZAi6qXjGbPiWAMVm2gTDJGWnDPmeyMIvWezj2UEt/kpd/a09f7X5FCQYUKQu9jTUjOAp+8lVeTZBzPWdVGoMrEwGuvSLxzcGs/gS7RbyMrwu3yGpWn25fYyXcDHtNaUUSA2zwHDrR2PvQB9cAB7CE1QIKEAFFy29GotBaaWUuVnkJl06XaUHQYh
J9+N17ULkDTXxtjkQ8G2X5cU4sKJ73EkBoV5DJsrNJT8AQlxCyDocsSwPHxy/xBIbOHPBHCRfjQyotn61JkhyKhPP/1PmFyjK3YPGTEapGIcMCKgoQY9ua9dtMwBw92XNu7rrRnXFIRrP5TsrK+NZzfGhp9yQ3eXWnxB2u8hbskDxMt0r1Fuky8fN1wfyCuFikNlOTGB7gL5+cC3ozdRhHY8ot0hw0bZHB0bmnPE+XrTE3egbdx45/5tT2lGPk9/53ygEc2LjWYeDHasPJUa3fkjUsPb/TB8OKjMSv91VyqeeO2/LH3YLjPOTCP8g6c9M/kr37Hk/hbF7lXuMhfucaDIodpdVehBDIqBf57/S6pUXufpxn7YxerIyw9bqf2fdQfd96OT4Eb3xY9u4a4ZNbtv4NCIpcbj9qo0E0wLfUjfwUaCl9yCMHdB1jSpeV9sM7rPgyoIDtp0zyOymqIc6AM3JdeEuapKSNY0Zp6nmFPCLrpLlHJmcVU7dq1WK8d3I64tvbXgxGePvJ1XS3+WB54DIQpY0K71IKewQFI6pIlJVFz+n77/GRu6dmE5M0b1nEaT6yI8M28PQDnQzvqwAL/1HS/l9ZfbcAIWo4z2E3t3pkQbC/rDb0MOZc7jmnzvLSMa6jyq2lhXsaD4UpiZwFG/vvxQIpuJ/zsqzNMf5e7u8VotBTrB7t11Dr4zmBbRl1LJ5p+CeKOZZTx+IAfF0G0jA9mcOwz7Z/lhcDdMUmRUIo0DjJkXEQ0R88BJ6mnsPqgN3wX0LvnMpf4Fzvs6lnfJ3FmuBRX1zxqtxtFPNTgcxjhP99RB+7+mD8nwzA9lG5K0m3ZFgNa8YUBNvXsEK/AaqzqM6ZvL6smWhqbU4JuU6iYNbvyjzMf/9Rmmd9VQy2Lu3YHjtDsrYsBB6+WEw5BGVlRSzG62ixHTHGbsnYM6BQMlGsDRT/ZuMhh6WbR7Lr2QkzOaj0qBA5M28bmO5U6w99V/TmWALmEZ0RNwDO/65FXDZ3s8lSEA0BBjk44Ajza2Rq/wTv18k6Rc4C3e+5eG2yRl1Eqh5+HCbg4F+H2PNsZnzLXYz33FfOhyrPKLIRn6HazKbhPsHZ9qzJ/p1nmDA8+F0o9OKhuqfBzfpz21Tm/4WupBWYuqYdtS7QeAr5zKeymsl6QsSFTt9/8GEAVRubKcVRPMo2MKoSV68RYr+R4XE+GMm7ZcbU+EMxsC4zNF3ULf6ah3J27uyuktDtrP43X7jBlWOKKxDOYpYWOHx8F3aXGtGJY8YPwf34qI5bmP+UlWbvFrRbqvjYbUzbQThl5tzFyyeXtGC5j4q7kK0gyrYnV1DMFGkYfjdqcZOWyULbI8Z8T+3S6aRhnNBlgnbnamfOnBwTGd2pZlHnSn5UOEQcLcL2t0ZeYBh9bKk0qWXkTFe7GbBGjsvCAPgwrm+9IrWq9AqXi8eOcj8kXb9g1mkv9HOKxYtepsp6pmEVMVp6M7wwzoT4aBZUI3FtwyeqoplzjW0pQG1sn9XBOb+JpZRE3nkAkbVgJhq9Yp9dwSLhbWMV4hkgzNVsHR9S3P1Dn0/3BRrCk/ZFgTPHw3smPFUp1KBXPRPgSFnR2m9BXCQvtdYuPm8qYn+jm/WU4f/HuF5R1IlBRvsrE17LHbAH0njYnPCMeNz4VkIpfybD5KPOPQgB8bWg6XzH8/+qAjc9RwuA5VfOnEubE/9mVRTtr4GwT60RPpN5jmJzP0rtkAdRlqDXrqgTDdKJdFBhzdqJpxWuq2xDnl0O0bHyrl7mK3aNCZ6I/c8MBzgQi2N2vl7MFNm9klj3t8kiZg7XnFxc8TuPOrP/1NP31TXUUDFXMHtfzPmoTsKlbdoH7Pne3CFe0u4lL8YJ0dE7LPUmdPHvfRMVglbF7DzF79JSDg98aLBOTahdjn1Lh+k+8TlcRWFpHqJ2vbRmqK6UlEEkOGTK1byGdFG3GWBZB4oyHI9OefGfNcY/7y5k9M2LHZsQRtzY0vnt2ZZ1i91DN2z+wA1q3hzZ6UFJgFRgNmrqfpCHn0GSoX4NOxqE8mQj1xbt1wnRo+gnBRgZWfySXYFJ1ZR8aFqRHnwVbwp5xCV1IcoIxhKZQiF59G6kRIl2y6TOmaLhJTGpGlHFqtXpuSfKnqhW4qESYBLYU/wV3enVlhj0HxjYlRmzW2sPW9J7P1JrHJaMWf+s8uUJmKNRq3EBkNbmWpQnBXpoBPawW1aPc0LvEYY4mM0+RF7/Gsrm1D/cgDgf6MdpJ1mPEpIyp+1a8Vsrq3ySzfmZEOYojAPtpNSyx/zJpVfUa7WnTbf4BdXN5GiSbsvq8jpc8vWeFSfNrwe4q7f+Vm1BCX1CSjnWmrFYaiB9XEdzFxwKVHgWWcn+AwKFqApzzGaoPmMmfPQtXuDqv0MoHBTj/k4/zS8CSj0jPNyT+wGaFaf9++V3jcF3lvTyOe5O0yaA//7r/sWM7xCvDCd1H9rXkW1XI0qGGjhI4NPtPm5s3VJl2SGQnWpszoW7sOf/G6uRsIIfkbI0xijRaAmwPrb35aiE7FfHomQQftBAoLmcT/Ig8izm+E5OuzJNPYd2jCLRYSLmd+IgrMN8WeBpA7nU3kazBFwW1FJgQdnOMVm5i2o46YfDAJqpnsyq+tcs5ZV1WBmdmEZBUPPIgTFdwSR1c3wwZgxQnLRH3UEzItfvVxyuqk0NX73DTKJDM7t3i0arGgNPJ8m1YkDSw6C2Y7Kxxx5GFmxLXEcg1d6cWbUxHZiwBoYH2Pc2l5HUEmTrtSGvVmu8b9gcFSfMJZgMhFr4DKr0fFeyEPJBtVV2JpvAjbChERrEJDxm/Mvlx9jkzC/yJHwkBErlIB8mKnrQTP2qQ5Yn29g3+5WzCJJHJMUWNXTBfK9iHtkcjrZqWQqIpywcBGHdRQkh+E6pW5OlAM1cv+GyfUyClJKZKXxCu+VwGa1VlxP1sS0dOfVB8BCzPVsi26MX0QObEMoVmcQQj+3BTeDxVzzsRhbeuFwd3g0BqkJN5XGHDGkP2Z7NKZ2R6+XHtj1ZC3JvlV3S1SHxGHrUoHtO6p1h9Fc6Tk8gVNLEKFOQgJj7ZtC7AnMGXnu6nJBTWqCPWR1knPWBYCZYWHjDMytqADvoSPMFkTTWi2ncaXCBC8n+/X8vN84sTuO9OMIgOvo4yMMi36xZbm5nD54S6Iipdlj5TROVl/6K8IFc6VrMx5Gk8hvlUANL9PzjsEpOpMD6H8IJAfYwRlOwLsW4eRDjiq0utfdq8TB2NnrGMOGpgr+bgMv2HdmpNEDkfTkXgFKHhZWd3dpftyc2C9kOBqmOGsuAbl/ohoMbGNfBVUF0ZfOv8E3tVrIqnlrZ5dOkxpH3p1IEx8v5Fbj/iX4I5YJRhkqd+MZcIpEPptJwX3tC6D+VNAi7d0VLFB1W1KHFvjyMJbBxLbYzKKIIIVNkkcwZ5OaTGq4X7BbqFZ0Bcnpv+jS6KBHVaYdXlVqFEEVQXq4D7NLuoqqP9YqaF/3QqiNJ7OB5YQO6gfXYVmhyTWp0sG6lDE3Y44Ba7kaM9IbpDiaP8f1kfuLmHQN5jCoeVHsH3xLzqUJ6uX3BQg7xoeQ4E0G4C8wJMIPNIJoBSpFY8OcygA5pR3uyWcwfcSqHrTCcs5gJuv+ZF66TqP2O4ju4N+mm3rtQQOS0GyFeW5hYdJFR7MNhcPygQfuKEBrEvCg4aBwvqHfxbjLXzrfIEsp3g71gkKz+7m6XyBggXV+XBdKebDN5/WiynmtkmNTeVa0RJrQjgdONYT6VSAUhaVFhtwMbyoPhDGqh/8W8f4FxpTmpgjzO1AeDX9rzmGClt1m8TOo/Lv+FcS8+NQpwHqMKI4a20EIS9L6ITwPjYy3IwWiVUEHsWaSvR4KXnI4hKIDvH8PsmO+I4Fzj0oiyg4Wou0J0EOy/LFPSdw95cW9l8gtEMyAbd1MC1XpSh4hgm2qKhJljZ/Iu+Tyw3dE3ChV5HOpIrqswkzEl7GXfiHSg3Y06z6Ajrsfk/hQgd6y+zAwiCV/1+XM29Sbxnfwcbi0SCl6fXmlJP1jRh7MsL2H4u3pQeL0Ko8G/EoZ/8vFrlvZGNTFv9OdLEYfTq9cWNR/r3l1lVVvFPc/tIy6dMXLkfJblOexfGTWXeA0PZqbRmoO5wxQ6ssf7aoRfVoqH9gSHRqN5fq/RMrXM5g3IBE3KsA9WmM/+yvOiDZ3jQkqLm3OPFJG1eSJ3+0Gyp48QjIMAkC7/DxJgk0VC4jBdWNrtxzjgOW70Qpyc0oqhHGhlYGeukWfK+TggIvD+x8V3rLrEIaOjODNuLBYUAJrMxchzlAtGmv+9aSmttK3Le94hX9OeqB8jU/VDjdydpb4u4Mbw+PA4b8nYkNeKD9v60BNsLX1m+BgxTeDtBJ0BbjjuIDyuy8mLhYNHzFDx6/pwvs8Bgrpfxc5YTuB5xn/MC4tGnu6i5xTC5gmDM61wSnR6b7Ijf+ZqXMjLfI8iCZqxUtdG8JAZ1vgzYXCRmcANpFCaXtXTcMNVPrf6i3IsP2ZeW5jcX/5PkQAVIZNw8RM4iYHijnnvEvi6xQB+dt0vUCsOb1hJKxa5J2SBNH+Fpsl3S7xmjZhkl+0SC0k/Lh8vWb70bwHnZMwhqzspvoNsJ/GU4ZZNVF8hMqgzA9TgqStvL7Wmg0C/o5gjl+HDnL30oVkwr/Lp4tuWdeyGgfFbgQe+5QZk+jgo7TUxukLljvi5t8RPpQfpm5NwQY1oFr+xVxrMvAJ4ZTWkBvCNR2E4cIJ8l3Py3aTTdlfCdkpLmzqL+aHSg8bZ9Wn3wWIN1rIa7L/jFbNOmi3WeMcPqIhrKaFUeRvx5a26H/fDH0Ic1xRxVzcH6Xl7B1GOqMJY6AwgB5+0whb0uYrqy8+PWR8YBTP6/TIqL/H8ByIeddNMFbbw1hZP+C57aGKP6AQbpCCzMUSs3R9f4nIYD8UEqIXE0L4Yrt5+N5fUYcqdZXbqtA+KO5o7WYUd15UAGTgmPyrbVvJSRx+73yomBH6sSjW4+5UO6/X9tUc3x+o7RZw4hxfnRM6G/TUwX5yPDlMCsTtOeddAbDi1BEzVhIbUPGdjJlxnGHJ00pNx94J3oDD8SkRxcV4DERBVuD2fw4eUtfgBBxMuTAlV0nWgMfqmJZAyYHiHHUAbfzGmr73VScVGJL3YjGZlW1RoRR3RNHQxAi/WUe0uYYdDDjvIXQHZJljJP3T9d2SATsHf+7mWxhKgjQTQ4/TCrHPNrmDM7XANmQTNQmRzFQ1yZ0tlJmzBe7RQ3cpdqlgqCBqxJkBTu8iYaFI+SLmNn9tzVZGMTMrU+FX2DIQ0/Tb+6k5ZQJGm5oWTFPCSQ9jb7edCpe7GtoRQZ0pM7NOc4RwMQmg+HuGSWU6Vq/R0yrGAIMl5Y6BN3xjxYVpvSmRzNFbl3Lx/6Rf6fLPOzKz9AfrWQzkPVpdUz2Pt/Io7k+m2FvhWdHEt6BE4GQTOwn+Jm9ev1sxhVp3m/VgjmCJxW8/Dt/BGVhJEl0WJYEDqWNM4lN2pHKObIvyd3BGJmTIS2kxRUf1GHtzLWtBbBuN7i9xkAJqvJ+ZKO1eA8u/YGpbxBVe03NeIuPwMXyjsLg+MB4iTsuYg4FEeGvY+HCJy4x6wN/XP90D1uCG6ZK+WuqVF3hPdpSvItL/1gMwWy6NhhI4uBmuvSzklK8DOfHdDGaQkQJaRqdbIYrEHl9ZEijXCj2xPSn85JSNAjq5Nqrm9cQ6I7QaplnMfhRLeGRtgnCGhRQH0AYe3NdAZ0WuqbOuKf3Nrj0djRDeN1MEdKGdX1Dj3N6ZDDQtyNApC3mXwxoHO4/JXzHFtLTc8O3WCb0L6T78E8Z9aXYw02Jm8kUZg8k8Jv+RaPCDpkLRIAiX6CeNxCESSrQB8sFCv91MhOYW3bWI+yhnT+G1TKcMxy4z+ORRpdI8swNxr/yvswIp0HZ6BG6d+9ApPxpsaUuc6+xMUzQf6JJ18OXpc3mxntAkNwftZk6rKYK6dZJGh0dyoRn5BSrcsoJAw7oKjqAwPLqnjAhsygmOt6gYQf2afPXp5ZeOVXDtvyO9ebuMmnODn6JP1vCuSXkeWPEX0fY8OzKuE1XSm9whwuVRPEY4nsoTkk0IfBFzhMeOF4dE21HdLibTJfGx5eP7wyZK2L3WOXBoTwhfbXhZ1cg4UOBN/eP9zyNAbrqNUqcf9zGsIqu6+OcIEOE5IcE4zVqxjb2aSaDqNnL3K+UWjRGcMYqn5V5tSwvAqBf6sNjRO0gGnfz6ToiwP4YRg78nGj+WCkeYHOQ6P4MqT3p+gf1P9MZGk0HcpDX9RLRWsAxLoT+FjaQGZmMwS1NYh8v+Gd5Z8Sx2IZ0eMpqvFHeJ03MRLIQZ2O5W9Bw6W7n5bKmM8+Gfh8mDZZj7OHemX9Gm6RG1aEKuG4LMlG5RhvBzyH0Z+6TmNlXpp5W/1W8u6EVcZcXzh2vcHpp9n/02Pxn/Yvx/hqoYh2dAD8H9lgTszDN1UxrUZaMvw5DSI4B/Kgjw7DQU/qSrXUyE1bPyqoGLOzZr8/hgrgrMqU3lKmd2yG9lrM5Ao3ePbXL4dqdW5LBhsSYjbO2Fm2xhC0S2Qo01aWA69jPA2fAD4K7UX5fBlRHYm3/0M8KhnaQnU4oGEvPSLAjS8/0/N5gSt/TF5H/1bvHJOVV9uhyxsp8Ogk0FNyQH1zQAB8G+vHcg/Jn3QmH2B3efsk3AcChpZ2b/SmAzfTwnkfZk/JxjQBQ8sbhRv/D3dN6sBQjMzlOfXXU2E6XcHMTkrb0tHrAf0Okd5Mn3gFWqe1+oVFOYoFXG1vDxjsLjFPXL0EH1KqJ1y2mkLcAcyP4aQ8KwlDGY+9gb/Iz8R1t7cQRMn/CrXLy4cbwKCHzF9uG4D1Wq0FTTy5RB2Sd3OhIu9yyqGqgR/pa92Gvt7KtJzNPuJAIUSzBh54zNx5PBfBi1AoCMB1IAcbPQPaPIP6cfD4A5C5uFzX4uhCW3LMDCyE0fEUBqrdUTw5t9XFWONoEiq8ooeSXzMVAWupHK3OIpeqNhIdS7brNoUyDcy7OxR7O/YoSv/zoh1ktjj3Kaz+V6kIUqeJCgo5+eyqWk2PJ7Ic3l7bXV8Rkcd9ckeUVtGyKb5SExhq2F98d7m8AFrv3r3HU8e5O+ux2h6D3gwSIkPjtrttBdmpYRSbwIkRLnVSLORHF+eIpxGFM9gA8I8GDDT1E9iWdY9BgxRdkdqjKAyvQRlp6qNTkzFbTqxkAOiNGzI66JNm4SGpkiAmhgt/EW1XFoS7WhFy2Qy+MGuU96WeIYmDU7r8GkOJOHMdJGdTQZjmuAOG1H4QlhfEEaR3NYmvx4qB0qYv7hZDVEYb61HCEVrjdi0w5LX2UAvHmkbRsOHYbRwcpmPvwzMHDbUd3KXtm0xt5WN4tvXOW0VoGLxM8URGqjvKOTszhp9kwxY4h+k4XYmqyXRzTF/ew/l1Kyj7sa6RbLSeZYV7Fn2LrKt7ur6+D/XugXLDSuQgSqtTxrQd2HHrIQm3TzZQ7qkNhgNLQiLDW8hB5ijORItQUYh8NJzEfj6AbIVlgbMGNsVw8j4GECs2wUgnAg3OsKbHkI41ZdZBbWJGYO8fK2wmiVxMb5dmgiNmO4P6SfYDGJ5wi6J2s8LV17lxXNNAhUXx4O0XZMhEDgQGU1oJcxRuBI1o5YnaklIMQ1JO9L6BQ8tFyBq5wVsV3HcqnFD0Fs/4mB+v1qXHM8rYeXLypOa1ByvqXxKlaEWgz61TzvLk4TW82jXHJcz5OmKnoyVc4sqvaQnOjZW/3ZrJmwa5n+j4Tt9yRR0+Eoi1wjbQYFTnTmyyTUlWL4GfZO24mF0bgTGZB5Ke/f7mgXfkDkqzrRAoX3bctqROoi4w2LT3NLnz2b1ZKDF8DvDNZ6u2Wh7Z+FmB+OLKtX3QPoRH/t+KHjRFVMuECt0+MKFsK9L07I3xDqePNqOMpX/l5GkXwyzaVL3tkJ9LZoqJNSZzP9h2SLCnHBtjOlL9vvVozYFx5wPmHqzXfSOq/DqpqFwKBPeR5X58TKPyl7PK5RbeSHIB/PQXPXLRZ/6Ll3p1mW0r3XPJ3sVXhC+rKlyDQBKLFzKYT9L9qcNvUwYW84bjhiCfDFkwJcC7FQZr+Z3zBAX6hNL4H0QiqBqyhx0BMJnvI4SUwiYE+Ahyd4LfPl/OugJZZGyamfo5JNA36d/ov1mL5UryYz51fpJXHoQMr8jjCk9XzKJWmnRW+0I5/KP17DOMZx81GVbeebHXA64a0rEowLe/QgSgXKaMTtN/6g5RMpWzI9FEz3XcHCzhk3VcVYy8j7S6UEOKdhrEaqFKXr+uA07GVw1OfcqSfNJ/6ULbEl6qEIQlN/4mJr7X2s3BUsOj4bVMOWs/Xf2K7vU4Tr1p0vrDuGIoE3MYRYRRwYBBJ6SwWJOvhKDmtRA1LDZx7VXpbiKXlJ/RXELI9bZJ1yBGYAp8jRUNVl1Pjfz6mEFpeakVZvk8LcyvoqXyRmkRSYiT2YP2c73x7N3UIqsIsrVI5/LGOz6b3dGE98oeWM9I3NhlrjrFn5mSnsAOrjfh5SBVEsbcEC1e87fAfBZ6kDmQIkH0JEIp3JL2tVwu6fFGNrOPVjsXcPfWGM+C8E1p4+32xqzPLNo/U9mKKJAqsqVD8gPh9TCX+TJ7aB+JqyCFnIt2TIpnR+QHhDgrb4hnZQ9JyiNOUu7J4ntkTi4cuYh8HDpLl6QTPtNgW/j+hbGBk6IGxbjwd3TjqAQKQeSNIrIiQOaHh7aXrm9HTcsWVa6y4gytv4Dl8AhpLAsTiuxbDcwkrf/KiiOUmQ8mD5hsy5qnH4iRHd+ttbLXc8qt3KNvQp9tZK3ieKcFPImPygFgVqupYiVNB2S4oAXDTJy+1tXuykxiCJwEBIWAG1ma8fhS8xIhIhqqzjiiUv5pIyl7o1FzhSm2UPcN9yKFEFtqnX0DtCYyrpw82C5T6Od2MNE9nXmYtzk1RyvBuQtn4vNSITmKBeu3Pqk/3j13e8vBB9QcK1RtCEP8hqrpdfS/F3uuZNPX1YyhnqqlylfExR5DtDp+HCUTxbhf8blZsP1yqxyc0M0k6JmcyGlQNcQv/WFbeY76pKPE7HcBYhj+Zfhm4zV3UMS6jyMNsYbR9+sl1Zm7fn68TyE3Pkc6n8qy/vFEHZyCaN+31ibOCdrjBDMdAfnB3uNZc0etmmqz0S+qXlA9fK6mRGdvnnCI0xP4vXmn9iJZ28D8uPXq8zJzuNZaOt3BgSIWkGU1MJV0SuFL7vLItWrmqtZXY4rm76M3xrfMDiC2NDEfSa7uhsKfSGWXw5nYWH8cUVvb3JfKDE8VMxoWVJOMxuD78ft7C9VFhrvjmM6b3Ak3WIjyGCpGWXMxzBUGmUwiRZV38fXqMYm9PbvlA6tPKKpDrlfvI2tPmkyJUGrubxR0bEMz4z4m6hAX0D0jHDmKcmufBUDWdzSY2twdXKpX4ySM8Cz2yBeMDjz0Iu0g7YdwHl5Z5y44mI6rUT0qfNxyGh6k/tmfSSfbtB1MpNQTd6DhO6ZCcPrw8iJ2bKrnUAhYBtAF1xYqy9KTqpP+mmxMIXVB72xqUHiZ4YXxl6HApX9X1PWjldnV7eYsSTPYI9mCz9g4Yx8QaSpqLCtHXwKCbRHkgKL2NqbHHVeQYdLD8gmTmrZ8bvHx58b6ufIXYKuvTz9Lj6P3NPQFIgCkh+s0kiWZHk89D0Hgat6vD48v4h6V8SNIwgNRsQ+tkrM8hv9LkfL3Yu0JKz5O2VZC9pEg31bL1MAnGMn8fdPzjGXzuR7Cod9SQ6pVbHqPVsoqD2Vdbas/n97ZPEkH/9WWVB9MGppBtfvx2uPuS4nvwZ6IwuFt+GVbgbbzdGdjMAl0yYG0tRfMFTO7T98wshMStk6zXJ3MfnXXQ/DkUCFTs5GrN+oljO6m5rJBA5pkUVbwg9/fuuQGvfHi0bD1d8rr0jJHS5HFPKNxOgkhyAlHdbFnZ5BMRVtAPdxB4GWWC/18/IFPWEb2+fbn3XMj3BBiCvyQvu2ZavBmedPnIcBN6NDlpDP8n+HiVaZGq3KuEW+7wkVPMFY+O81JL9VdKhEg4S7LzDjj2nS0yXEkCAAQV2zdSm8EV3oKvz+B0R2LKZLlmW9sSIzORtuwcS9qbqTBPq8a4pt+BbFS0NfD6JyyVk6JqZlsKjNPtaTVfunhr/a+SRYL57OYx9UNSEnOYpkNk6CMGxwbKgyP3+6DQRDqxmGzWHPhaJTtJSqZ6rt9+wVxxJH9z+UNMfmzWJ18zwAhuTBfHwHq9XBsozLaB1JdrDOkhhNyEqT9FoQWubJIeGZ6sk1EWEhzuFIIKt1qZoyr85ojr9FbH56zvJVUJq/7kHW+RXyNgGwNl4Cj/ekmAjnJJ66RzBDuBK8m/lX6ehXm+pOQVtP+rxlyRu3yTj3Grp31yW1aA9Q8KXxm5BDZyRNvJz4XljJYGpQBK8yhJHEYF34N6xWzyAuF4hx0/mhvRbKkn8J7KhbHBUf59oOt/Kk+ELZNANLU8iAA5zWNuRAzV/aXTxXJMimOwhJbOwdPyHMw1C8WBSXTeuQbqAbNpu1HdElqE2Xo98BaqdVXd3+/VJWw8ZDyJnGqqvN99qP/3w6M1i0Gi94DxQI8vqIKCWSXDr3uqc90W4oFAn/VdB9dj1ZqVlOkNR8B5TuTEsV6oAl7JwPvnbJ5UWKxFsUcPRCeyOJ8/8CoMJ0MVKhhx+lsrRTYfc3QJz9C7tKe9sgDqiq2jA7bpJBl0JiHwi2S1tsbgvHsxYUr4xSEsFhOK0p2BXn1YQyJw0nGBmvmCsLt1vMB/XwkuY+L5MOCJ0ugLX3gNHyZxfojR24c3S+8DdmvzD5RR2JbQ4cVdhrq8OoLIEhm4y1+MgwKIV6PzkcbhnWUGeN3vEfiALlKVH6XPGp5sOdQlypmFGuUGuNcKVJhcTOd6VpKCP8BhDyUqxgTF2lfpfl01CEDPvfDzMGl8qQDQAbqy3CSZSpdbfG/v+GWr4emSnU30JAtekYE3pdIlORNsg3kggX/vaMnr9XvlJ8XRmOq+PZgJ+PEgYUAWOwuSIX6Bef4EE3tpEHz8KTvCk9PWavV0PnqeA3CVCiKSrQJfCBIMpHnI8rAbZvQKaxtSbv49A8jjIuwnI8sYNjxGf0o7mQKdhtsXfNrnzfI3KkdMXjn7NVpu4r4NDzE+E9HTZqEb8hwUfZw8Yzt4U2nXDm/2RlW7Q/vs8kUukaB2srEH9Atte8Q7IYFGN2hJTY0HIGn7J2twHdcLzF2q7YcMEHkD1f/NXQFf9I7SCPIEm4fnEA3UnBVzrZiKwnkODhcn9xZxdRvx5C9ylRrZmWohmRNJHLYU4CoQc+SDvwP4jm8tJ0ZGD+vtKXyhH9xkhq0YyAuo9uL+ALlJ1Pzr9EN8hNzrSLnpdt6lv4qDqBcTc8jxr9M5+Jp1bKyGMa7ttpv3jAmsQn/JvGvU1nqGwOpmIcfUAJX8LHq5frAgcGHeZyCx6NKH6yKGsA+UvR0iaxrHzWFg3/PgDtd60IxHwf5oWo0VJ3wvnbs5YlXkUiwRuPoYx2MTVbx4rGl2drQLF4vrW4f4iTrG0yUVFwIWo6w+fAtzwcNfhhXS2rN8k29fPqJz+JTZrr7krI3kf5L6WTnM0JGZOm4rC+1NeyPYi06ByHYMd5AI5zYNxsiV2zQjuZtebTT9W6UwRRT5HRJcMgp9QSh8tV2ADGHb0WSKUnG/Zza9sxyhRAFiQeTeYKVR8jNqvZP14I79yOOKoa7PdTNUK3dzHyGfCeVa56OHO6Ab85+OKBHL5XtzL+JH4ckV3EnI3f3QCY9JZeCXuc3rT7GB40rBqU1dD2NrzyA0k8PNqMkUUKi6SRGPdYo1O2yE/5VsrsyD7uWK3+Mo6Wf7M8GXKsZfrWfBK8RgfvqzgQoc1GmL7mBm/u+pvHy3f/4DuWkXZgPcJrhcu7j4pLlvN54hpAO0Eux4c4JqHiGBBqQq/gqKkFbC4HYFozFeHGxvY89ozgFPzUmWRMYmWFqLVfyQQPvzIhtB56jwFivgrbnCwYMCw6yHVMWJW0/lSA5PYZUmPPCqUY7yiGaUrbPqOVNMv2n/L0UdJZ3mkA5WyZ1jH7Rpn8ljyWi44Qmvz3kwSz6Qq5fYbCtUHZnL6RqApAJUwVjUIE48ieaNZEt5Vc9sEItlkhZF7eLMA5mCfTcZUNH/B8Szxj98FXm2HdR3Mv9gzv8MSEVtGXZDxUKCgVtZJLN0VhaHwL6KuoP7GcUqTW4O47B1a55BTZgH2H9Tao16ZsF36o6RHNNjT3wPTGUPoKHRjw4cyAxCzcfVOm+YtnbTKMOZzJN78FaXqP9ZRfGiymNPVKpAYZ4fmpoBa1pddBmYV5nV1nPz80T4Iq1b9o7ui2SgfW/HJ3WNnI1Ib1W1NWQqqBahRNm6RTzXUWzCOVB3tAwNRNKlHhQmbe09J2lyMjKP7Bsy+VbH8iZbvQBi+MBCu3UNrsgd8SwRhmS6Nl/3OxNsdJ5BVzLOxgQ0iJLmywYV23jsIRNLDS16NqGLU6HX5Zi4KW4xhQgbZHN6jkiZzRqubJ1gjBqv8u6rFa9SA0cVBCR8tDl6Nj1NPCjZzDuuW91Y3Nx8IVEHuFGWQcIvX6ngj1OsKUOmP1m97pY01nOQv/2bPbMJGZQlYW5tmgX2UTOJ9B7wyhaA2wdwBZxmO2tlvUTLNoCUUz1ojBGYZNu/ddeDe8bN1/ajDGrxB0CRhNfYS9CjaeeKDb5ueoYOeKJLRYsIo7NHpfSQEJsLmEeT7EXIjlTuaxfXq9yW+Ri3LR0RKZwjpv+oXPfWd4oeFSoNoVsxcIfqhDnGrYlBYk/l4I4uu+T7rtSPFQYZyBd+xdbGCU/K8v7fdePCoG8WFIvbOUaOt+X0HHqIpuhoUjz3KvvnJNTjbLnfF6cg2nfjV1h+mGSPD9svkFhgfj4pUxfN7GZBTjCuNEk+O/jpRgT/MtdVo9PWp/3HxXL+O+ddRyDP3fzeFvYq8xeT2lDr9RSxwJbn1/6Mmv4YVnktWnvfD0SaXqpeWkrvg+cUOW/wsbB6CXDBxyNjw6FW+iHmtuyvMpT5tiC2bojPgoo08LqqWmlvdvqB3BqFAsdPhCEbDmBXEkyCQ6IH2VJwadwMsVRgP6ZdgsmdhYIPVmzBTDujnYFJdRL07G6YJbvmBXQS4Pe1sjH90xk1D3xTK7L6hKi/p+j1nCOGJRYL4ojJmkbBlQ0gCZFB8BEnzpFqrSbwiiHr5VXfcrM6SfDTGzbBuv5tL0CDttfWTiR5pv7Cqf9v2Flo+4hyRmqfIWvoE9PVD5QQGFuY2J9sCzu5C3t4uoDRqNrcQgYaJW4jmCpXKqLBiJbcLw+3bTkljbqPQSfB11w8dZ9rKuoEaHNeDXtSEnPOjGgZGfNw6nDX3u58yRkjHbT6x6fQMHqEHa89MO+AkeJpVwBJxb7rCMuJ4OTRBSqNkzN1vbAV2xnkEnqtjP+Zr4nPwnwcXW56qt59JYSVGfRxw3fqKQKgLBcZXQLyh1fZ6wzk8VCcZ2sW4tJhkiaeY6dc9vgUIddAr7EiYl1r36a7IPkuc9iXT6vj5lqDyIyzK4P0Tqtxrml6kRKEyV5Wm4KSWkL4zsa4dA7oKIpzX78SkWtjU1AcJxXE9lFqvyHx4pfpKSLH9OEIFFtTOTBHmMsq26nafi2u30WxBSDQERjsF0j9KAaSTHzOXlE5nHuaYXxYT4XPyzN2HeYwxIUE8aZTqTvc0wQdqzsxNFppt11Nhr8q5wJG8xO9Apw5HhKWWB9gHHRWZ+FFrg5gnKvXjjpfZ0cL3P92pQQSKcv3LbVsP8qIwc4JBgx2A9YBE6O5xX2BkVnjTjaP/IaHVxC4nGqGjM7VrLcXtQeOtLrZzTsgLb/zyGhlMdKIzXrWR+sDsfS3Msto8PmSZWwqrs7wZRx/M3ySWyM/A3Lu6H+GXcAKpOgYhHFs/zQMI1GR1jqbv7l7hoYuE26PdMr+Flv/cMKw+lBJgLAdGgLRlh1Y2hyQhqOtO7PFr3LJaL7Nuyb3dRyxmZvBHo1mZCjnPFn4pBcbfwrl6IJwGb1aLmkOeJPp36AKj3Gr/dVzZD45tL+yrVtZcfm0oS2g1RQh+rXN6d+X1h5O8GdrYf+sbfh9gWrRphlpGj5Vv4mC7Ggxjn0tCGKXbCqz9uF4qvLCX6F1nhKg0XTVqYmYHqS4PrGyhhcsOyI0Vq8uvMrLiL6w/AW4jSnRjrzPyLqaF/Wf+QmH3Ml0eR8e4v6yk8gbGrDtUcE1VzqIahvk05XOVGljOXl+Gh2LScRqkJexYO8wyY9hsaPDBCbGNV4Cyfaa1FgnaP6LrvYaMWzwsOL8WxVvjVIh1gBq35Bdm/fNbtzLUmpMCFTLyKlTS0dSYxJJmnaVZ0pXOxHz5uG9ebG2wtddfyZnqzzk2cJXHc3g2Vmc9ptDH19YDdo2rYGoqTbqv1SUGSYP7vM4t6/FjnL0Q0I+Vhq5N0QDUzcrIffS2zQQ2IyO5Ipm/6SZRQdKMNMhgK1JxeFTW33GbvGf6rKnZVWaAUtX1zsTlwqzJH6vpKqQMIt5fBVewDUXWH4KhG2Y+FGcMvxb1PxsvBXYeN+8SZ/hEuwHV1uxPTy4m3X/xG0mQubjpRSRcrIjlGXaefYxTILhrHhh+2t4DJvOOuOwQj6AX3VtvwZe7eeGp2+zs2GHSDOPZA3YgQIXBa/WfVnqd8B0Sjg3GdW2CjKFdv7YlqU+JSC8oPSRHTjYBNd7ZpDWxNFJ8IX7hbiq5m/5HhMBOPN2CnqtHw5nO/4RsGGtiOk5PjD4Z9B3U6b6wAJiT2nyz6FcIuwixRS+hk0l9TyfxWvNhs6wbGU6k4OZGrW+fGiq3TeL1JfP71ZDhx8GjRGiw+5+9sSX5DuSsfOXpVfLXdPFc75tcKy9NTbpWnYwKn1EeSTdQEm6XdI3xLqtg5cUyJVf3iGfJMVLmJH2o5L0W3/Hv/wQvefsi/x7OvV/vNxNvL+t5cEa+eU3LqXbU/rxFIvfijW+N+8DuVmK5Z8zyfbCkc4Kf/tDkH9CEPcy6PAqq1fl0Fsf7Ggp6ig9owkgLuB7VyLcd0sutCbkLR3XeOWOtLfCZzwBMTGmPqCq4Mfe6o6FXsEGsFzDg9BpkijgE2dnEDYRcEvvD5uSGgkEfiijIDMsdKYttL1MKXMyz3N7JEqQddMWEjBTPuwshvy3MbhDwj/ShH0fOhB0s2NQSwfQDnMXBGhK79JQE4uZMFhJ4eG9+DdXmD4bjBKBK+4vLlnV9K0BBKNK2omwAfL8ehlc5qHW79n8zR0HqYaJHiLKXK3sVmnQaklWkLEKwMFzQH84Rfzp5eMM/B0TUgvhL4z81PbsvXzlXIA0xmp9LrYYkTh3z1kcUP2aVwGmMBmMGmHuJh4rWp/0sMlnBXMQZdqj2GdKHeG8+s1+aWMjFRfs7/DCNbeRVoYJUwBx/JrS7CXy8zRpAo1DW9Z37xaAe7cVvzL5pkw41+RQ4z/NwmJ8ngqH8SgUM6g/q4AuvTFj0nNjNGjoSu7PYkNqp7V8bxeSvIptD/79wrR8tm/v2vwLvyEx3v5Es2AQk8aKLJ4M77KobRR3fMWs2gQOLpBKrEhaq+jdjGV9lL4vlj3jBiAJd3eiaJvWjhQo3dpl+Ocz12h1Ax5A9UjGXVT7cmOhHv3WbCNtVC/3EIyMuhbtW9SlB2BRPsFVz6iNySF5Ms5YZhPlnFRkkT3Yzq3KmeEJN3wwvbzYIIGdtwhkBqhtNtwf6pHdaa5rCAm1BZAefHmKc90eBRf+wb9cVYqPx7Mwg6Th6T1H4CPldzuSmNGoQ8JgnwDoZFTW2A8w3CfSDq6SnNHJGjWdADwoADljSbhaQTdTuQ3pMO+Mj+xJMKjQ6r5yK0ftpzR56tydnufEpCGrTKrIhhkpq+63NqG7pZ3DjHxGqDp0iJ6CBq/vVRqg2ExTw1JtMSNGj2dCAyZeVvc9HKIaC3KDmhh6XA3tSLdOjLbpYvnt16NCrHNcGGhotdzI6kXEWrl+C5fMRS2aGaqxPaAR2GVsJ7VC4kK4ETtT8qE4SpJBZvpTPMxtzy2nGfZVyI1ItsfsJodN8MUrX62pgQD1W5HEQVcRmKOUhG1JwEl1Sd36yQ1Kk1JR5u7Ttsvwva0O3FQB8P/8HOgwWqRGbJPl3A0BELtRvPtUnqfhrvGpe0fKQ1/tWHBF0xoE6zjOPfu1Sy9q2LrtSFGZhsRXvGAHR4EHADv6kk/Vfbsr5kTrj01Lz2rVSdyTY4vj1xK31H7m7q4IoWhMoXJ7mtD1aHKVQ2h/SBjCEm4KGCqzyq9LUqfw8bPCqWa2XGXpi0Jmx/y7GdJH7D/aC3FV+XBybldvK8rb6NUNpa/4I5YZzr+tUDZCpAkX6tFwa6UyAL3pBvyuqdYp+GOyKvS0CUyeVfTnr5nEFzsN1MfM5d7UFYbkAm+4KgpWsMkIeoKMWvZzrfe3jbDwt4e1DAX0Y1FuMOjFMGJ2iJd0ODwcaSNZzCV4iB+zG5bDvg5b1gWsL2x9xQE/aH+bxg5dN/h5F4EnYYz5AO0NSOkN7cHqF3zSWoJGAvWWBbKfHdl6rDYtNC3lS5cpP+OHB0JXMoVDK8XrzGY3MmS2b9M2Z3mKlRG3gy2zb7cfyPf9Vmzh8fG7rwAkUz5gWn/iSBM4Jv+8ke+Xtbk0NjZ9NJbP6OJA21pX1MkLBsx0oIjRcX5kaP5UmPNjtG0iKAmjfv5QtL2mdDlJdPzVvr8rI8GQs5vCFUf7NEOZnJiYeyG9yKrOqG8u0zr7vh14QIM3m+0n5xvHzgF8gwvBc43NZTN5CPmAD5yt1NUttl44CA7/ynnE0/8c2mC+ifYS7EdBuPfb9TvDc9WIR96UYJSBOf2kcsmtsBEvE8t5AmQgE3bt4hFnbumdm0r3JszrR8wkmQ4UIEoNmz0ZsUfBdydrPz7VYcB7B0bgbeMpTd6mIxyb6fPrANvc8G0IcU98bTxDvCwRiQPJ2TRfRPS9nKGgfn1YTi+BrkIQnEr6qpNXOu0eCl18JPGAYSyYvqnrHUTu0eXc1FSa0dKw7z64BN5qHvYeNl/RBdqIfI8TbaBMbYtDMTIO+EqpobXOqoSbQT8iyGL6xBDzg41RHA0I7RL7lt5lqdhxzOwm0FyGmpCGPi6CPzsDlHpHpVgKKXfrLv6rC9c/wkKPyBY4dUG9wCOKfZA7gnpbKRQhj4mwvE5R8hxgbM6qmCHY3IwK0Bqm4h/9E2vj4U9nXn4iPIyMgftxUIm0EVXeqGmQa7hFmGIWOQbmWsfDEDyGDbv8DjtFiWoIVWb0xd9ehyWTXCs2XAGfnObd1NZ2Ojl4k0aafhXp98R6yimjmpnzklEPqsU9NFfcGlrTDn7b1wqGxR61pUktP0FTBb0mchuiCqKtuIqIcsf9HZm6VG3v2vU4UZRh/hLbsQqe+L7lsU9TWeatxG3NYqFdMhHhZ0tHBU0D3DnBzBnSMw7gjM04T1xkTgHXxQl2lIv7gQMXYmrz9uDIcVXc3wzpsFzP7Z0fy7nrFfLk7DY2BRuV4XdBDMmjB38iUHKfTMvvFtK6Tf3u1fbn0Qc7n7NimEvnYUb/M0RvzdljYbvSqg5D/cXgJp6FmlgzJhR97cE/EP8E5OYjmVgzyRtE9obBZmxHOMN03LZ7tLQsXVjDvEzFYU4A6Ffo4a4qa9N3Uvlq7Pj4kjWpWYlGz2eQRebY4iCM7t/q9LcVsrdK/+FChYgzJAXoVJqOeBr+AXaYRxmd6s8SGQlt0w91rPBt5Gh1o4omqg4z0oKGWl52FmJcQDqQQwb3g2x0eJnFmmWHFhAkrkW3tjqCLPE6Zxog9NkJ5HN0UfjC4Ibs9TR8lLyf5OTHpX7NkUhuQJZw+Ak/CEn+mjC0khYljhQu+PX8qOyjVmoz5HwSxeMPHJV4kvkjhYV4Vo1O4BN+lpc1mbuw4V93O98iN+vpo0WrhJXzrSrh+pXOhL2AtuUORYJ4N9HMJd9LxVV8TKJEiK8KuVOr2R9+EIPs3WtvhxZCZB6zMVpeTGODMeYfuWBQuzW6Rgbqrw6DOtz93/7EF0cuAqO4ZHNuGkk0VBlVPwLp51o1uy8p+xSyXRuifszzs6jCFTm/c/PrqUIiLH6AnPUIOFxBjUvi6QK9IBC2Og4ksUIeFDxHUAsGSbB3fFYtFTOBDmw5gjmrgh01K1xYuZ9Tb/m3fUc+Eq+eblr8ZkdF1nbxTMZX5qpCDVdiPEY6q6Bbc1U7HrWz6BmvRLAQMSRQB+xvAJOf8Z2kp0zybwyW6DMaDmcjGoGzVq4LWmVtfkbkr1Wpyx1s/FW+QJDnlJfsWt7mxYKUqGsCiDThPGoWGW5Mm2LPV/cpdxvn3r3rygr+uPofFd4bVWq84I7G3Vrm9FrzvhWrTSuXNyA7Jwoy6S3yEkR3Z2nnjh6nLfjZFqSjDU4wpxrAJEUvH1mCPCjnGkbDvLRrjbZ1EPkp8EpM8hGvg0cgdrJZvdJ8FqUOtxIkRZrSkfyLQPqGT5V1hJDmmxHNwMGbCzfv9Nfcb7xCIkpu52QUCkaHTRzvJMQRQlt86p1+uZGBUoFtBYKu16PlnMq2WfrITEYEdOj3LNM1aHNOCgzHtkGJ0Sj2ztwPJP2iPuqR/bO8gqqGaAU3vy9AZ/XiWrS0fMRuMQFuSwiamHJcm3kjbAC9S5D0YtTMUNnVjYuJ8Vad2/q43pSzdTCOWBqdmhKnHY9hxnMFsN1NNqdIXrM17O6GwUZfVNKxnb/5c5BW+rs3dlJpwTngqjD0M1qumgNXjJVw0/ejJAzIXH4Qq76jtzjLtw2RHBujf6wo+Br5Gjvh9UpnDmj1ahZ9sfKKOKivKRmADBa8lijA+bqq3NT5Ad+nwRiTqgcQ3fqn3v5V4rIVlSdOtWwhfyz6N4DqkNgpcJjxSefCZc7mKkgGR2S5xBVPbzMYkvmr6UnOS+sTfwjG5QxSbW7LtN1WV/9EAJmXFOu2cnI7Nb2CmI7ljRTE8kmTRYHY9IIH0Oxqmnh8W/rvz3c5LKV3p2vUmuTplcA0jRJhRjNd3PlMfZkqqjfvPyYaxrwXCdkWVhGzOW9ZPcqBx5rPBcZjfkR2Fr77lmTw0b7fFuLdoxmJkWZv6ZE3+GimxfKC2HoZ8uxVaGLqGnBoDkK1ddZqLr9CYKPV8nbTyubSWDnVv/Pho1p2cdmrwnR/31PQ+OL40UUYwgwAPU4pG/x8Yg7MW+eQdA1MH2iBIkL4DkPGcTJfP10OGHFKAEJx9wABMD+58Ky3X1bVAEl1jyx7Nz8pBWpdsknkTBWI6+DSRvnAnoJHjcQSdZmIVzFcyTcu2/6o9DmxFmaEvYTyKdO3IrVpz5ThgqotbUSqRpX7n2P4MGrgLA0P4GmZotaLuK1e59zA6rZgpzUx9m/ey8NwlhV1C6emifC5XOWySXfQCmK6KmVYRp2adYvyi0HCUzMqQUfoJH9QBlxLjHDRQ47NIi8aLZfqS4lN7rS2cKvBf6WRC5fxu6W5Y4ypzPqXr5q9Cj8nJL5yltjBTxtgZr8WNnqZ40G4l5Dnd3xShu5NkbaWM9OqyGoKqZ1R9iB6qvZqHmmKbNVj3V64Gus4Gr3I/tc+hNeAMJcGfeuwUwxE0sd1kHBP0z3Cvk9/ZrX5uT0epiR4ddV+LoEXFyLP+rZ8zmdz1ltEQXH4IqhQ9kaeuUQZwaxeTUpGqFg+Wu/x3kA0RC9L3a+E2wWtvRo4CSqrcc+7YDpDYoPxEAD+TUQpm0FQRVIkoKm+qXD8KlB/GrmSK61Nz5fu9z/JZBwxUtKO6SaFzMG6hFbtbeD39xBlq8GnRgd2zJYFBJx/5Lr9IkvjtXyWz6fA1xwFc2K1UGSRt7dvI/aB6jXiSX5bcEtuEK1kKEsoz0wF6rrCnj7l5JY7E3HMvgaQeUUzsvCaEmyDdjuPdZ44wE8MlDWvpvkX3r2qt4nP/WVhJkqVQpN3e9slKdpqsLaAHgNNp0p4LPawR6hlW1ol6p5b3lWqbYvK0gfUNwRR9sFLR6XR8+fJRhRd0ut3vJ4fJjjSHfxb1lVk5ojtMlxRRFX1zVsnGjUd+7j3ESeC3SHkcz0SrYpsAPDfODwNPJL1nEUVfhkjI5uVKylfqsbM+/tCgkhI9pIJyCX/D9p4URM5/VqZHjwpRWxMUMzk2JaH+oG/9ZLqmuTWFCkjoyb1wZmx4AEbK2PVjnxvhMmgHlyx8Sx3PqgkJHg34oIWMbEM1bNpLo23YJrF8rjrL7TmV3+B4xIFPRgbnqobTOmClAnA4hdtdL2bcQ1nO+ikZ9eKTSLkF+Wf8hWsVVd2Uj9YNhLT6jc5S5xsspNdA1tz9W34UzE56Sxd8jVmr6ZFUtDhJZjIlJ97nXUSQ1VoDMh4JbNLXzTtTfCn8v+QqK6tMm/FWmmtCZSLCxeNrf9gH9PhIxQpF9qtM/hhYQZbQXLFdVGfn/BYHbGQW9eZIeb0Xq0M8JD3ofKtwvgsDAWbnUjF8dWmOEtKGhCM5hrSTJtnrhI8A4RV5GfVcS0mmx2v53I3LGuH0mmJBXEJ4WFQc9HqA2hy0fYtdJt7hqGW6N4Q2qWTnqsPqmCMr5ESp/gcV+2OsQgaq0cZAgFWTdjtghgqPWkFeLLj0HO8XdTvA3R4jCKlmIEIq0YYr5G0i+7Mwfnlp8z4pqmShzyJYbT4H+E1XzklbY0Xhpx9vkHpmp+FdoMGmHzUzWoSuszhSSQ4n6bvi5RzlHbcS6HBAqzNF2EiGqhNtDmDoAtVdN+t8RaqtdlfEJxU+2K64X6VrWHVAQLKrbKeGk/DDBWOhMoCxftPv6bjk7O/m2mDbcwHWpy9BbbPu+8f5S1lNTZaGW80U+EPeaMnfnRkhI9MGdyo6ToLv4jUvnuGPUKEqLjppKT6cyQb5D6lvEHkxsTnqVrHVHy+EZ3OWuwGF52lfVwB3rWLbJQVdMdD68hdeHmVcP7Opnz0mzrWu2eFsTwFfKtV0FW0qcu8GZwGbwUvB7GZb6+jr3UGoATbMrLJTrHgPIc8ePUYz9A5VdfBcIwYsuPg7Z/2fdmhPpNNSOeT2iDmP9EbTBMXCfXmO6LnDZEjqgGdnbndbNhSKh/H7g/bNQtVn9RB37R4yjCLTmIvYuSbcmHzrpFP5EMNgHmS+OOF9Dh9NSsdJhS1IPSOj75vq2WrmQ4+GoGPYSBpFfTPmgzXAyRB4HLIRgp2bPdLbV2kQ4NDhrfkNQGC3LUiJ7lXoYTnmQwt9MltY7nvxSy/3xPm8vCqdvl8ypLXpB+CBGyaPeV4a5Ec7ESYTbu8lYsm0fdnjvAWcz1rEyEFL3xk1H1e+6ePmVHcT5VjyRuIuYQzx5/d9Gz9wnlf/WfVc+lJZomYB7AOVuYGWpcA7MNIfCFf2rvGNY/NGPI8uVvZ2ydD4NoZfxVMPYv1cwC9SC3eLYrYczI4Gl6UvbGo5LTEDtLsAkZHV6OhNyk3+ZxV2zXMfCJDedFa71W3FAD5bfdYctVNeWgvV2a5V+CJyjTBMuI+9k3xRNIYV7TmP+zyKpG6emBHlMJRiENnQoTEf+lj9fQvoN0+U2V26zuAklPkvEZmzYtBpXJ3j7rRXAKwf0t4lfGvOEHjnfI7C04ikEvQLwIR/NhhE2xjFRHhwBQfYAgqJxTW5q9rnRnJ61/8OoTXCGZNRoh4iJqbP0WnR2q7l4BhH6EU1sTmM6ZUmQEyPLJuhpSPbqT5vvBJz+j5VFWeVGmMpNGIrL+VpnIQkbbd+TdhtM1V+VxQb0wUehCSyWSt9mCNKXyGkxX66oXj9eNPDe2GpuDkoDNpGdy76DyGq3jWK9aq+Fi+3NmmtdCGLzNTXTqU6ALh3ycoP/5fMNUFD3sr1b0P/kT9GbRdv9gmBcjCzqeM6aVTTnHMdEJiR6nSjJfXy8bq0jMxCwPC3o1Eed2XjWNyJBzKnDjfWWdKQmZHkjq4iSLWj6SEOc3HfyxsEv2B6MB9LgOPhv+k7GBqnRPdfvzGzWG7KLvGmXGmdh8CxC37VvuhQP1I2HbgTYCMwPd9GObpz1gjHwcHL4LAK0VT7XMsvYG6EOtJGYp0Lnh/QTNLJL1lNLJ2eqt6MpPtcSKbXIzHuYOgqSWS3CfhZUXUkqdWnQIZfFg1XlXzcq/a1Osiic0Hr9VyxLZoG4NebhRA7vl3ZMGSMG1Ehs2yo4ooFowYPoQ8gqMU3VoprOjA8r2jMcyyXX9V6NAzmW7C7N5GiUQbvNqIqzWcBv5xSzPjWSc52p7lH5ByGTd6YXF2gBTSk8tmj0FfYIylC+RibPzx8UwJDHdN/rTh/WBJZaq4yoDX2YKGuF0CbgEJXRc1nHLdgFl2WMG9nS+LVVfhTqw04vFTyaN9Z9Sdr7tqDEf5+Ee51zyDA1fb1b19NOo8g7ItzsJHIgn63MeiNwFBZWeAMJ8nZ2YwcVX4L+u/k7AGt7Rmak4xaaV4Olz3bD4/tSxAfAiOvs2i0rhkbCVyBmgZS+CmCjmjfE+PfLQpKpk1aDmHpsGZUdnhNRLQ8VqRUXRYIZwX6eAzQwOYNwfLZSJSBfhrqqmsMo7ZLzMAVnJgl3fN1EtDE0ubB4T6kg6I3ytYO0QfygFU20yI3sT49fc3o4MB2PpCoqaQzLMuxEIJBlIjy0dlMiqx5K6yPUOKZlAbHE79zo9sDlfT3Mq/Vy4rBeyf7SVsdrWvF82iS2+KrGDv3f72PyfTjIWQMgrRbL1j8EpBtHmFl39v/rnwTQG8lnV/Hfq0FSR+/MoSr1PkdOyvX+hXiU78WyEIEb3xzQOgn1SaF87p8A8GhaeSUyrMTj5v/tCeLp0a0uKQDjXG8sTixrjt44tTeqAYBuUoGCX8CEYDw5k2tFhZMGoiVdKKxoLB6tQjwfvR7RGxINwgFiq5gJLeHodsDlt2GzGrClWV0hEP+Rt1Itfbe3nvp3EUPB2Us+V/AiXCjQE3JN9fCnq0hBcbxdRUBQRoUdpAh1s3qJuf+RbGRyLGJi+JZuBddhHoSVSL0E7bShDoSA8oTFTOdxUpVD+2hZnpkPJTQ4lBk8gpgaXgt4V2ZcbEDnEFgM2rj0Wq8jjIL2Z9zaqLOEDi3xd4bBwkbR5xjRaIS6uBQEJqhQ1smGY5bp3oGt/wPa8V4XBqWOkSZfh3/Df0gk8pc+JLc6CMWWkdwkP/iRhrCb6a7fqbmW1yUdZmE8xdzgjf7vyzhZ9kWHdeq5Bz1pvFUj0PVPySE5jIhPbu7mSmbJ0D0zHL0kt9FpR7+cPw3DHrXNhbrBMZxVeQe7h8DqW5iBkNzhPhS8Qy4uX28Fv2cmVIj3+mqM5RvmNK7kJ85gMuTzXIJbmNXdDPfkE/zmfNN5CKWN8HqHT2PUFEsmDdt0RUV1lEmx6eL6nhyFU17ilJdqf8J57Oq8JFDbOqBOENmg56Jc/S/CsE86w8IvbEWMQl5fbsVjlWiJC4emNxnUwXrLBnQOoNJZ06QumxKQjkJBDfwLm3m+8xlmLzYSGJQ4+oJeS7hLbBzAi2EmcGzMQ8Bl0kKv04KNfYCUUFXB7k8fNlIlN6whnZa0cVjVvKpvEQQcyVlFK6y+VmTRqXv4zSs3qdXKtfQlEtfAbDiR2JPFI7oyF0mZ61OjZWn9SgtvAVhMCm1kJp/aSBV3KrsWsM+nO3ZaR1rXCEpWl1bM0uU7JWzM0TX0QV8mPr1bC8Nmn1e4m/vtMaVpVYIbIMkDHDzPGaTN8sywOxpApBJLhChkZCEPl9DR9Q7ruX0Pq2N5Q8OwnHkWYb5cCM8Y7/IiDOfrMAgTwmkeJNugQg2klhh5jdGXVXp7bojIuuQ4Ub7rlj/Ee4f3cyg8q1y6f/WD2yCD4EtODDn9PEYhDEjrbi3zxAijmSebPfDhqv1SxknlEY+rTyeIaP5hrw0hndYuSb7BS4eTB6cu71OjGsQ4zDsunJDUFJMhmugQwIi9mqnHqpMZwHXnQfsAAEeUqfoJnpHRc8qIszyT4xONVjbkkrFLiQMh2fOMCT1MmmGUKoEm6dSYBiBIQmCFRwPeRFOsJaFK0SuY4mOf3VSFXAMvjVnBeq6ToxvTnG/Dfq2tVJqV1qHvVq4Y6bhyWc2WFfd7pmBjiqrBdsdgpf4EvnTRNIm7JtlIaGGyEOhXGzwfeKGfqpYnNLEQMNlkXe1EOiJYjXGFk+j5nX1J+yeZbwjXB6pqcghmItrPVgdKiaFvBoYM+r/IL9n1iRCwZkNm55B4rpQ4h4nmf0E6pUx3wcKbvTyJJx3oi1pw4hRH8FK8rUvn1ZUVg6G9CPDjhGFGX2v4JwDPnsV3nC8E8UgdvUwKSnOzKnCwMtRRDAnvjJl5zqVdYbKqV9xOt+MD4AaWthfY4uEh1pYt1OAKiyAvMZ2KUNP/GhL5rLW6ItqQjtG5BKuTTzqjpdLDvbLV7pD1MH9d6z1Lg6GoFuGfzplDJY6EgcAt0L5CrO/Kp1w7rCNfpALqpzYzSWpefKRxo2phK3mvdOFPGALn7NRUJ0PhDE6r3lIc9LcO0SUT4vNkJlWCD5g9vAcWmXNC1h+sO1UHB4ThkXV5qKpBCnqdVt+ZzPBeJmMNf0txmUNUIR1XXe/qmNT9Bf8ZbxKzZT3HSHF5SCqPOhJXrjcHnoAuPJYXUGhJdbuhCH7J7Xt3+A8Y6MfPy9OA4XgGP+bAE9ff/IaRDkX/s3mYjL/KIkQJ+42flJFYlPwHaWkH4/iJNMJ1RJIWa+qGC4MZdmEZ5UnDqRJ3xBNDvyeUmMQzdABHa3xM4h0wprIkUnEN3sFqmb8iHD4ULjnE3f38Z3xHOMqaCALg1Jxy6uQ6V0xVSnEBUBq3xAKLXQZSJ5wHUi7KhcTGhewq34vjf8Xs5jb+yIHbm4nTmHsEiVeeHH6VbzE2YXFDzt/XcQWKWNOrPvbGFPH8sN1fxqT6K1AEcWyb2CJ4qMGYrtivPp37fPUYA6RvHnNPQHn+MEdcPTnlgIgwWTBGo8XP6VoELa5zgupIN/jYUmuapoObCHmMlzIAqVyv9nOctMl6UQ5u61N+hSkDpEaMM6w7RO45WeNRa0yq0XMVCvIcjq1W1DyvU1OeFVZh1oz2uv9W2wKjpqK7CSDnZ5W9pZVTLzPM/hK2gjA91C+KPdaLCtgBcaNk2YIj1PGhKWdnGnOCT8q/SFgVDXGLGaTbYsDm3fU3KClBKWwbOjlbpwBy1WZe0qphJ12ipJTDdK+uGSKeJ3hAXCJpxvV5qtnjMQJGJwhSqYvFi8NzyFjD0x3OX9pG8XmpNy7s10ENl0Im/JCCrgcJrD3FZ1v4lOCSpONYohtwt0W5O3EhN+DGQ9p1XKNWo0V3fahgBcwPGszBrBoUo+8tqjLWbODT7TJSYX13ZcOAaHu8t5gqJvkE4lUiwM9JmuV9gKVF6tjqnvJ82yXlZfQJ0Y09nnbYVwcXwHnzTtyuUdtbaQd/4C13Df13tXkfvyGRPmHb0DuGiDC9i7Y2YSAv1CjoK1YjkAUZN9h1ELb/rIX8uPmh+eAhgvLMrLCypFzhVvOp2Pbhq1PDylrpVbQNmwqaEVj5ohhWZpnPdwVw4wmYhu1W3/GsJX5M9eCu03zYyCrTbBePgNrA4c8Lt6HFYy8N7vySOlWSwbu/QYAZPA8oEnFjjTHm2UPDxP8hpmfxiQ/0xW4ucDmg5hTf7m8Crh4dUMeONw9JzNbhV4Z8TSyr13qiJDDQ4CbO1FxXFR5ikVRtk0FuUkn2IQYyvsNLFC0H2dn0P3vYQcSDiNZQD4+sf8rOuDxGSBoHjeX4PExK9Xq6PifyXVpcN0ug/3Jb5fKWp9C+Ao3hlDmQ+97vKgs5xi1rDQQQhGh3Sh+yMy9q2wpHCUVBo9i/ItgeNTFQ3YE31uyPDyY5snmMDpuvyAFVQtb4gSUN7ORLIOv12rRDE/9YUnbIFaMvZd/p0iS7twPuR9TJkMhSlfpTLfNZb5+8Hw26cpiDtV7DYKUVhoYOR/rfVCaEk93YIPNYnmtANp9lKBAN0YVySbdlRhHvV+2IjBMOj3U6T2MpgbDSu7GfmmaMTXNJ0gBK5gx28HDRmKBUCLpPukLfN3BnpKtlIKqj5hMBjtlQVr46EaP5aR2nn8Cp4wGVzVjLzNQaoJLgcCR4Um6mYrDkqRnqrvlzg+9eH5csF5iHKTdZNtgoY5ZlTUIgPspStjx9C5Z7lI3fP1UZvErKtQsgp9j2kMmqJw6ruuKy8CR0U8pN5xYlaxh2eItCUOh8/YsFVh+3qnPXPjVa6r1VWIPkZ/JGNkA6pCVRwJqR2BBhsPtj37gGB36mWHnGfRP9a+uJKJa9JxhBDKw5UDzGBJTEFi1q29MwcuB+Pi9DwAgtNo4RuLZOSLnCiGZ3bqJeUmwFJDnLpEXkZHSfsMBoa2SDd0IKZzcHVUqv5wsRcHIXZaTZsJVDVWU2tuhwsBI1HwPwhTgm6IQDpqgVXqxNvH0vbhH1MZBeGKDw0JlaASRD3rH6C/+b6YKR1mWu3tDbXCB58GRp2J6QTmIgD/KzrX5h3APyiM3wl0EhGYyTt7uLX8PfIA6llQqHQM/cRZ5lzXrTHYxs+cK8p9CtxUWrjOs15hzDl/SxE7VuO0ik3tzIxkRIwYzkFsvvMUB8EQ9+ZHUoK5E7CuxTbNT/SQbMJhwRAcGEj13TfyqWDnkQMBAjJlFmShtvgh4Mm4Vu3GpaTJzZ0CFOadEunijPlUJMpD2thRcpcGgNVqWtUmskL9dmftCWkNDwehqLyAi9LfxCLzvxfNdJ4R6hIgHEZ4WR8AU3NYmcAP8NWc4omNZiv6QE45yyOE3uqYs1gQE0fT3cJyEzD6hgjXT3mw2s7JOKNmTpE1MnYI2vCOSv7gqBpWlrbrFjTiSUPqs3yUDBfkpmOuBbGZjmn6kapIzNSMX2pP7EmzhJRZOUEjR45kz549KKUoK6v5DutBgwbh7u6Ou7s7jRo1Ij8/n5CQkCrHdOvWzVbWpUsXcnNz8fHxoVWrVrbENC

Tk3n++ecpKyvD1dWV8ePHExcXR0ZGBhUVFRQXF7Njxw5ef/111q5dS1BQEMePH7/s27plyxa2b99OeHg4A
DNnziQgIICzZ8/StWtXhg4ditlsZuzYsaxevZrw8HCOHz+Ok5MTI0aMYNasWUyYMIGlS5diMpkIDg6+7GsKIYT
4lbOTon/7RvS/zGImx4rP893eY6z+6RgnSkq5rVNjOjZtgI+7C6fPlXOs+Dyrfirgo+9ymLZ6P2AZhW7SwIOyCjP
nyipo7u/JrId61Gn8hkiSw4O86dUmyN5hCFFvLky5uJAkz5gxA4D//ve/TJs2jfLyco4cOcLOnTtrTZLXrFnD4MG
D8fLyAiApKcm2b/v27bz44oucPHmS4uLiKlM7apKdnU14eDjt2rUDYOTIkUyZMsWWJA8ZMgSAmJgY5s2bd
9H5paWlLFq0iHfffRdfX1+6d+9OZmYmiYmJLF++nLS0NACcnZ3x8/MjLS2N4cOHExRk+TsPCAi47HvWrVs3W
4IMMHnyZDIyMgA4ePAge/bsoaCggH79+tmOu3DdMWPGkJyczIQJE5g5cyajR4++7OsJIYS4OkE+7iR3aU5yl
+a1HjO6dzhFZ8tYt+8Y+wrOkHvsDL+cPo+7ixMers409/es87gMkSSP7h3O6N7hlz9QiGt1iRHf+pScnMzEiRPZ
smULJSUlxMTEkJOTwzvvvMOmTZvw9/dn1KhRnDt37qquP2rUKObPn4/JZOLjjz9m5cqV1xSvu7vlLmpnZ2fK
yy/+WiwzM5OTJ08SGRkJQElJCZ6eniQmJv6m13FxccFsttwoYjabbVNSALy9f32u58qVK1m6dCnr1q3Dy8uL+
Pj4S75XoaGhNG7cmOXLl7Nx40ajL2MthBAOwc/TlYG3NL1urydzFIRwAD4+PvTv358xY8aQkpICwKlTp/D29s
bPz4/8/Hy+/vrrS16jX79+zJ8/n7Nnz3L69GkWLlxo23f69GmaNm1KWVlZlYTQ19eX06dPX3St9u3bk5uby969
ewH49NNPiYuLu+L6pKenM336dHJzc8nNzSUnJ4clS5ZQUlJCQkCQkKC7akdFRUVFBUVceuttzJnzhwKCwsBbNM
twsLC2Lx5MwALFiygrKysxtcrKirC398fLy8vdu/ezfr16wHo0aMHq1evJicnp8p1AR566CFGjBjB8OHDcXZ2vLu
yhRBCXBtJkoVwECkpKXz//fe2JNlkMhEVFUWHDh2477776N279yXPj46OJjn4O5p577sFkMnHnnXfStWtX276//v
WvdO/end69e9OhQwdb+b333svbb79NVFQU+/bts5V7eHjw0UcfMXz4cCIjI3FycuLRx+9onqUlJSwePFiBg
0aZCvz9vamT58+LFy4kEmTJrFixQoiIyOJi586dRERE8MILLLxAXF4fJZOKpp54CYOzYsaxatQqTycS6deuqjB5
XNnDgQMrLy+nYy+nYsSOpqan06GGZtxYcHMy0adMYSxcYM4ZPAAx8uyHzNBfgLBjpOJe+65sckHtVCCCFuUkprbe8YqoiNj
dVZWVn2DkMIm127dtGxY0d7h2ByGxhDjNzYsYlh5UZzuXXzZf0/87DNAhiwhyxpnrRcdYmCXPE5JLx6awYlHjTfDU
Qgh6tKbb77J1KlTZS6yEELCxGS6hRBCVJOamsqBAwfo06eP06ePvUMRQghhJ5IkCyGEEEIIUY0kyUIIIYQQQlQjSbIQQgghhBDVSJIshBBCCCFENZIkCyGEEEIUc0Ntyy1UqoAO
HAVpwYBx+o4nBuJ1M+xSf0c22+pX0utdXB9BnOjucp+Wz4zjk3q59ikfr+qtc++4ZLkq6WUyqpt7W0jkPo5Nq
mfYzN6/ezB6O+p1M+xSf0cW13VT6ZbCCGEEEIIUY0kyUIIIYQQQlRjpCR5mr0DqGdSP8cm9XNsRq+fPRj9PZ
X6OTapn2Ork/oZZk6yEEIIIYQQdcVII8lCCCGEEELUCUMkyUqpgUqpbKXUXqVUqr3juRZKqVCl1Aql1E6l1A6l
1B+t5QFKqSVKqT3W//rbO9ZroZRyVkptVUp9ad0OV0ptsLbh50opN3vHeLWUUg2VUnOVUruVUruUj2N1
H5KqYnWz+Z2pVS6UsrDkdtPKTVTKfWLUmp7pbIa20tZtTZTLbW8welVLT9IndcRuqz4ebot43cZ4Ox+22j9dlw/f
pth0+SlVLOwBTgTqATkKKU6mTfqK5JOP43FqfVGCZ1rotsMy67cj+COyqtP134B9a6zbACeAPdo
mqbkwCFmutOwAmLPU0RPsppZoD44FYrfUtgDNwL47dfh8DA6uV1V1dZedwJtrT8PA1OvU4yGYcA+G26Oftv
IfTYYtN82aJ8N16vf1lo79A/QE8istP0c8Jy946rD+v0PuA3IBppay5Ay5oC2faO7RrqFGL9AN8KfAkoLA/9dqmpTR
3pB/ADcrDO969Uboj2A5oDB4EAwMXafnc4evsBYcD2y7UX8AGQUtNx8nPF77Wh+2xrnQzVbxu5z7bGb9h
+26h9tjXueu+3HX4kmV8/ABfkWcscnlIqDIgCNgCNtdZHrLuOAo3tFdc1+hPwPGC2bgcCJ7XW5dZtR2/DcKAA+Mg6pWG6Usqbm6D9tNaHgXeAn4EjQBGwmZur/aD29rrpPnPsSlNa3bjmaOH6lkMtmVRrQ+sgzqVuoCUJewl0ZUN7e/wVtmH3rKg+Xnug8CDnAFrWH
GT9anK6Usobg7Sf1voQ8A7wM3AEKAI2Y5z2u6C29jJsf3MdGfo9NGi//R7G7bPBwP32TdRnQz3020ZIkg1JK
eUDfAFM0FqfqrxPW/4p4p5JCPJVFKJQK/aK032zuWeuICRANTtdZRwBmqfUXn4O3nDyRj+d1eOwk+Z9KM8Cbi7/yMhR
Hbi9xfRmx374J+mwwcL99M/bZUHftZYQk+RAQWmk7xFrms6SRrlg62lla63nW4nyl63nW4nylVFPr/qbAL/aK7xr1BpK
UUrnAbCxf300CGiqlXKzHOHIb5gfF5WusN1u25WDpfo7TfACBHa12gtS4D5mFp05uh/Sqrrb1uuv7mOjLkezJA55mFpU6O03wW1tZfh+hs7MOR7
aOB+2+h9Nhi7375Z+myoh37bCEnyJqCt9U5NNywT0hRswT0hfYOaarppRSwAxgl9b63Uq7FgAjrb+PxDLnzeForZ/T
WodorcOwtNVyrfX9wApgmPUwR67fUeCgUqq9tSgB2IlB2Il/LV3Y9lFJe1s/qhfoZov0qqa29FgAPWu+W7gE
UVfp6T1wZQ/XZYOx+2+h9Nhi+375Z+myoj37b3hOv62jy9l3AT8A+4AV7x3ONAdtemdemD5SuCH4Bt1p+7sMwB
WwbsAZYCAfaOtQ7qGg98af29FbAR2AvMddztHd811KsLkGVtw/lAw2cPrBjB7xhnwKuDty+wHpWObq
lWEZUfpDbe2F5YalKda+5kcsd4zbvQ6O9mOkPttan5ui3zZqn22tj2H7baP12dY6XZd+W1bcE0IIIYQQohojTL
cQQgghhBCiTkmSLIQQQQghRDWSJAshhBBCCFGNJMlCCCGEEEJUI0myEEIIIYQQ1UiSLIQQQgghRDWSJAsh
hBBCCFGNJMlCCCGEEEJU8/8B4fho5HxS5n0AAAAASUVORK5CYII=\n",

"text/plain": [

```
    "<Figure size 864x432 with 2 Axes>"
   ]
  },
  "metadata": {
   "needs_background": "light"
  },
  "output_type": "display_data"
 }
],
"source": [
 "acc = history.history['accuracy']\n",
 "val_acc = history.history['val_accuracy']\n",
 "\n",
 "loss = history.history['loss']\n",
 "val_loss = history.history['val_loss']\n",
 "\n",
 "epochs_range = range(100)\n",
 "\n",
 "plt.figure(figsize=(12, 6))\n",
 "plt.subplot(1, 2, 1)\n",
 "plt.plot(epochs_range, acc, label='Training Accuracy')\n",
 "plt.plot(epochs_range, val_acc, label='Validation Accuracy')\n",
 "plt.legend(loc='lower right')\n",
 "plt.title('Training and Validation Accuracy')\n",
 "\n",
 "plt.subplot(1, 2, 2)\n",
 "plt.plot(epochs_range, loss, label='Training Loss')\n",
 "plt.plot(epochs_range, val_loss, label='Validation Loss')\n",
 "plt.legend(loc='upper right')\n",
   "plt.title('Training and Validation Loss')\n",
   "plt.show()"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 26,
  "id": "0f4a6d6b",
  "metadata": {
   "execution": {
    "iopub.execute_input": "2021-12-21T16:53:55.277899Z",
    "iopub.status.busy": "2021-12-21T16:53:55.277140Z",
    "iopub.status.idle": "2021-12-21T16:53:55.684759Z",
    "shell.execute_reply": "2021-12-21T16:53:55.684183Z",
    "shell.execute_reply.started": "2021-12-21T16:26:20.769667Z"
   },
   "papermill": {
    "duration": 1.110483,
    "end_time": "2021-12-21T16:53:55.684923",
    "exception": false,
    "start_time": "2021-12-21T16:53:54.574440",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
     "array([[0.07391414],\n",
     "       [0.02940544],\n",
     "       [0.07304674],\n",
     "       ...,\n",
     "       [0.01020581],\n",
     "       [0.00300625],\n",
     "       [0.00155538]], dtype=float32)"
    ]
   },
   "execution_count": 26,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "model.predict(X_test, batch_size=32)"
 ]
},
{
 "cell_type": "code",
 "execution_count": 27,
 "id": "5d5f6649",
 "metadata": {
  "execution": {
   "iopub.execute_input": "2021-12-21T16:53:57.097640Z",
   "iopub.status.busy": "2021-12-21T16:53:57.096944Z",
   "iopub.status.idle": "2021-12-21T16:53:57.215562Z",
   "shell.execute_reply": "2021-12-21T16:53:57.216119Z",
   "shell.execute_reply.started": "2021-12-21T16:47:33.831667Z"
```

```
    },
    "papermill": {
     "duration": 0.832166,
     "end_time": "2021-12-21T16:53:57.216334",
     "exception": false,
     "start_time": "2021-12-21T16:53:56.384168",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "Y_pred = model.predict(X_test)\n",
    "y_pred=[]\n",
    "for x in Y_pred:\n",
    "    if x>.5:\n",
    "        y_pred.append(1)\n",
    "    else:\n",
    "        y_pred.append(0)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 28,
   "id": "f9190dd3",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:53:58.642576Z",
     "iopub.status.busy": "2021-12-21T16:53:58.632649Z",
  "iopub.status.idle": "2021-12-21T16:53:58.645317Z",
  "shell.execute_reply": "2021-12-21T16:53:58.645838Z",
  "shell.execute_reply.started": "2021-12-21T16:47:40.933940Z"
 },
 "papermill": {
  "duration": 0.728792,
  "end_time": "2021-12-21T16:53:58.646011",
  "exception": false,
  "start_time": "2021-12-21T16:53:57.917219",
  "status": "completed"
 },
 "tags": []
},
"outputs": [
 {
  "data": {
   "text/plain": [
    "[0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 1,\n",
    " 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
```

" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",

```
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
```

" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",

```
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
       " 0,\n",
       " 1,\n",
       " ...]"
      ]
     },
     "execution_count": 28,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "y_pred"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 29,
   "id": "da9618f1",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:54:00.150409Z",
     "iopub.status.busy": "2021-12-21T16:54:00.149707Z",
     "iopub.status.idle": "2021-12-21T16:54:00.167312Z",
     "shell.execute_reply": "2021-12-21T16:54:00.167823Z",
     "shell.execute_reply.started": "2021-12-21T16:49:25.309418Z"
    },
    "papermill": {
     "duration": 0.815276,
     "end_time": "2021-12-21T16:54:00.167994",
  "exception": false,
  "start_time": "2021-12-21T16:53:59.352718",
  "status": "completed"
 },
 "tags": []
},
"outputs": [
 {
  "data": {
   "text/plain": [
    "[0,\n",
    " 0,\n",
    " 0,\n",
    " 1,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 1,\n",
    " 0,\n",
    " 1,\n",
    " 0,\n",
    " 1,\n",
    " 0,\n",
    " 0,\n",
    " 0,\n",
    " 1,\n",
    " 0,\n",
    " 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
```

" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",

```
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
```

" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",

```
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
```

" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",

```
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 1,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 1,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
" 0,\n",
  " 0,\n",
  " 1,\n",
  " 0,\n",
  " 1,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 1,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 1,\n",
  " 1,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " 0,\n",
  " ...]"
 ]
},
"execution_count": 29,
"metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "y_test=list(y_test)\n",
    "y_test"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 30,
   "id": "9ad5589f",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:54:01.602720Z",
     "iopub.status.busy": "2021-12-21T16:54:01.600459Z",
     "iopub.status.idle": "2021-12-21T16:54:01.609740Z",
     "shell.execute_reply": "2021-12-21T16:54:01.609059Z",
     "shell.execute_reply.started": "2021-12-21T16:49:39.348560Z"
    },
    "papermill": {
     "duration": 0.737052,
     "end_time": "2021-12-21T16:54:01.609885",
     "exception": false,
     "start_time": "2021-12-21T16:54:00.872833",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": [
    "from sklearn.metrics import confusion_matrix\n",
    "cm = confusion_matrix(y_test, y_pred)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 31,
   "id": "628fbfcd",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:54:03.041018Z",
     "iopub.status.busy": "2021-12-21T16:54:03.039869Z",
     "iopub.status.idle": "2021-12-21T16:54:03.046765Z",
     "shell.execute_reply": "2021-12-21T16:54:03.047307Z",
     "shell.execute_reply.started": "2021-12-21T16:49:40.750783Z"
    },
    "papermill": {
     "duration": 0.725383,
     "end_time": "2021-12-21T16:54:03.047486",
     "exception": false,
     "start_time": "2021-12-21T16:54:02.322103",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "array([[1538,   54],\n",
       "       [ 211,  197]])"
      ]
     },
     "execution_count": 31,
     "metadata": {},
     "output_type": "execute_result"
    }
   ],
   "source": [
    "cm"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 32,
   "id": "f2438e23",
   "metadata": {
    "execution": {
     "iopub.execute_input": "2021-12-21T16:54:04.488522Z",
     "iopub.status.busy": "2021-12-21T16:54:04.487800Z",
     "iopub.status.idle": "2021-12-21T16:54:04.723666Z",
     "shell.execute_reply": "2021-12-21T16:54:04.724131Z",
     "shell.execute_reply.started": "2021-12-21T16:49:42.548327Z"
    },
    "papermill": {
     "duration": 0.948203,
    "end_time": "2021-12-21T16:54:04.724369",
    "exception": false,
    "start_time": "2021-12-21T16:54:03.776166",
    "status": "completed"
   },
   "tags": []
  },
  "outputs": [
   {
    "data": {
     "image/png":
"iVBORw0KGgoAAAANSUhEUgAAATkAAAE9CAYAAABwcBXnAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90
bGliIHZlcnNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXBIWXMAAAsTAAALEw
EAmpwYAAAS40lEQVR4nO3debhd872A8ffrRCKIUqEqhhgyiBAkgmjV1FTUXCqh7U21FG1pFeUaUjQ1lHv10
tuE8lBKzUOCGtJeIYaIqHloEA0JMmicpC1Jzu/+cXbSk9Eue521z6z6z6/v53z3XrOk73Ymev7+Y877PWHiOlhCTlaq
WyB5Ckhk5SVkzcpKyZuQkZc3IScqakZOUtXZlD9BStOuYon2nssdQndp2i43KHkF16o03JjnTRjJjxY1rr6ilz7TnT
o8dWyx1CdGvf4pWWPoDq18w79lrvO01VJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTNyk
rJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTN
ykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJW
TNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJ
WTNykrJm5CRlzchJypqRk5S1dmUP8O9kxLDDGbRLb6bPaqTfIT8D4LTv7M0RBw1g+ntzABh26Z3c+/AL9Nty
Yy49YwgAETB8xN3c+cdnAPj+4bsx9MABpJR4ftJUjhp2LR98OL+cO6VW0WPzrnRavRMNDQ20a9eOcY9PW
LTu4v++iFNPPpEp06/bTuXPnEqesT4VGLiL2An4BNAC/TimdV+T+6t01ox5jxA0P8utzvrHY8kuu/SMXXzNms
WXPvzqVnQ+/gAULmliv8xo8fsOp3DX2OT7e/4RHPKlvlw76vHWvCsqwe8f+ONSE
ZsyZQpj7r+PDTfaqKSp6l9hp6sR0QD8EhgE9AKGRESvovobXFoyb+CqzZv+tqm3//o95LFjQBECH9iuTUDl1D
Ax07rExDw0p0XKU906bPLmRe1b+TT/whw8+9gIgoe5S6VeRjcv2BSSml11JKHwn/A/YvcH9t1tGDd2b8df
YtjhrNmp46Ll2/femCdvPo0JN/0nxw3/HQsWNDF3A/75ccfllAIy68w7fJzjvz/k7k7qcTp1Roign0HDWRA/
75ccfllAIy68w7WX78LW/fpU/J09a3IyHUBprS4/mZlmVq4/KaH6LXvT9hh8Hm8PeN9zjvhoEXqBt9hh8Hm8PeN9zjvhoEXrnnjuDfoePJz
Pfe0CTjpiIB3at2PNTh3ZZ9et2GKfYWw68DRW69ieIfv0X+VE
8ydnlz1a3Sv92dWIOCoiJkTEhDT/H3U9R+rendVIU1MipcSVtVPX+/Nl+9d+2MM5tVt+9x+Mx6ZnnwqZ
Ga8N4f585u4/Q9Ps2OfTUqYXK2pS5fm44N1112X/Q44kIfGPsgbk1+n+9Ni8K2+9+SY79d+Ot99+u+RJ60
+RkXsL2LDF9Q0qyxaxTUrospdQvpdQv2nVc8nVccnX21uu8xqLL++/ehxdenQbAxuuvTUND3pUAxuuvTUND8/+ejT67Fj02WY83ps
5kytuz6L/VJnRcZWUAduvfg5dff6f1B1ermTt3Lo2NjYt+vd/D2TnTYsuP3D/ffTttz1/mfouL0+azMuTJtNlgw14dPxE1ltvvZKn
rT9FPrv6BNAtIcjahOW6ca1m8h2SqqrtkFLijWmz+OGw6wC4Aw4SjxW2SnGNtO3KvcQ4bSmrT3r6gD/T85o0GmzqXazEC6fq+Ml8XvrGmabhuSjvaMh+T4DeN1OVVTOkm/oJuTLcsvvkwR4EnE3ENQsX/3Nh6GOahxpBuRW9BzUMhRZ6x8Hu8pwdYR
3BzJv/gKamhLH/+wGZv51LjP/OpfbHniKR6/7/0MfMXNPH0S29yxS3jSr5nKtK777zDoQcfCMD8BfM5dPBhDPz
SXiVP1XZEy2ftan7jEXsDF9P8Ep6bXVHNkXnDiAbGRpRUp/4kGSimTkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJ
ylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGT
lDUjJylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSstZueSsiohFIC69
```

W/k2VyymltEbBs0nSJ7bcyKWUOrXmIJJUhKpOVyPicxHxzcrlzhGxSbFjSVJtfGTkImIY8GPg1Mqi9sC1RQ4lSb
VSzZHcgcB+wFyAlNJUwFNZSW1CNZH7MKWUqDwJERGrFTuSJNVONZG7MSJGAmtGxJHAA8DlxY4lSbWx3
GdXF0opXRgRXwTeB7oDZ6aU7i98MkmqgY+MXMWzQEeaT1mfLW4cSaqtap5d/TYwHjgIOBh4LCKOKHow
SaqFao7kTgK2TSnNBIiItYFHgCuLHEySaqGaJx5mAo0trjdWlklS3VvRe1dPqFycBDweEXfQ/Jjc/sAzrTCbJH1iK
zpdXfiC31crPwvdUdw4klRbK3qD/lmtOYgkFeEjn3iIiHWAk4EtgVUWLk8p7V7gXJJUE9U88fBb4CVgE+AsYDL
wRIEzSVLNVBO5tVNKVwDzUkoPppSOADyKk9QmVPM6uXmVf6dFxJeBqcCnixtJkmqnmsj9NCI+BfwIuARYA
/hhoVNJUo1U8wb90ZWLs4Hdih1HkmprRS8GvoR/fpHNUlJKxxUykSTV0IqO5Ca02hQVvbtvyOgxF7X2btVG
zGz8oOwRVKfmNy33eGyFLwa+upBpJKkV+eXSkrJm5CRlzchJylo1nwzcPSLGRMRzletbR8TpxY8mSZ9cNUdy
l9P8xdLzAFJKzwCDixxKkmqlmsitmlIav8Sy+UUMI0m1Vk3kZkTEZvzzy6UPBqYVOpUk1Ug17139LnAZ0DMi3
gJeB75W6FSSVCPVvHf1NWDPiFgNWCml1PhRvyNJ9aKaTwY+c4nrAKSUzi5oJkmqmWpOV+e2uLwKsA/wYj
HjSFJtVXO6utg75iPiQuDewiaSpBr6OO94WBXYoNaDSFIRqnlM7ln++blyDcA6gI/HSWoTqnlMbp8Wl+cD76S
UfDGwpDZhhZGLiAbg3pRSz1aaR5JqaoWPyaWUFgAvR8RGrTSPJNVUNaerawHPR8R4WrycJKW0X2FTSVK
NVBO5MwqfQpIKUk3k9k4p/bjlgog4H3iwmJEkqXaqeZ3cF5exbFCtB5GkIqzoe1ePAY4FNo2IZ1qs6gSMK3o
wSaqFFZ2uXgfcA5wLnNJieWNKaVahU0lSjazoe1dnA7OBIa03jiTVlt/WJSlrRk5S1oycpKwZOUlZM3KSsmbkJ
GXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSs
mbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZ
M3KSsmbkSjL1rSkcuv9A9thpG/YcsC1XjrwUgLvuuIU9B2xL184deeapJxdt/96smRy6/0C22Ghtzjj5ByVNrdby
o+8dRZ/uG7LHgO0WLXvhuWfYb+AX2GPnvgwdchCN778PwK03Xc/AXfov+tlw7Y48/+zTZY1edwqLXERcGR
HvRsRzRe2jLWtoaMfpZ5/PmEf/xO33juU3V4zglZdepHvPLRl59Q3sMOBzi23focMqnHjqME4767ySJlZrOuS
wr3PtTXcutuyk44/h1GHnMGbck+z15f0Yccl/AXDQIUO4b+x47hs7nl+MuJKNNu7Kllv1KWPsulTkkdxVwF4F
3n6b9pn1PstWfbYFYPVOndi8W0/emfYW3Xr0ZLNu3ZfaftXVVmP7HXemQ4cOrT2qSrDjgM+z5lprLbbstUl/
ZscBnwdgl1334O5Rty/1e3fccgP7HXRIa4zYZhQWuZTSWGBWUbefkyl/mczzz/6Jbfr2L3sU1bHuPXtx792jAB
h9x61MnfrmUtuMu1m9j/o0NYera75mFzJ5s6Zw9FDh3Dm8AvptMYaZY+jOnbRJSP5zRUjGbTbNTsyZ08jKK
7dfbP3ECeNZpeOq9Oy1ZUkT1qd2ZQ8QEUcBRwF02WDDkqdpXfPmzePooYM54VM54ODBDNr3gLLHUZ3bvHsP
rrv1LqD51HXM/b9fbP2dt97EAV/5ahmj1bXSj+RSSpellPqPqllPp9eu11yh6n1aSUOPm477B5954ceezxZY+jNm
DG9HcBaGpq4hcXncvXh3570bqmpiZGG3XGLj8ctQ+lHcv+uJjz+CLfeeB09e/Vm0BeaH4s76fSz+fCDDxh2ygn
Mmjmdbw45kF69t+aa+aM+57+5RXHPzaLr33KLMu6GCfPfbX+fRcQ8xa+YM+m25GT8
65XTmzp3L1VeMAGDQPgdw6OH/sWj7xx55iPX2XXArkBn4B1gWErpihX9ztbb9
E2j//BIIfOo7WuIsidQvdp79wE8/dSTy/wLKexILqU0pKjblqRqlf6YnCQVychJypqRk5Q1Iycpa0ZOUtaMnKSsG
TlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKS
sGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMn
KSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUta
MnKSsGTlJWTNykrJm5CRlzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTlJWYuUUtkzLBIR04E3yp6jjnQGZpQ9hOqS
fxuL2ziltM6yVtRV5LS4iJiQUupX9hyqP/5tVM/TVUlZM3KSsmbk6ttlZQ+guuXfRpV8TE5S1jySk5Q1I1eHImKv
iHg5IiZFxCllz6P6ERFXRsS7EfFc2bO0FUauzkREA/BLYBDQCxgSEb3KnUp15Cpgr7KHaEuMXP3pD0xKKb2WU
voQ+B2wf8kzqU6klMYCs8qeoy0xcvWnCzClxfU3K8skfQxGTlLWjFz9eQvYsMX1DSrLJH0MRq7+PAF0i4hNIq
I9MBi4s+SZpDbLyNWZlNJ84HvAaGACcGNK6fmy51J9iLUFu8bWkd5CP6LcsbAu2eZDNRkjEE6NbmX7CTbDNCvnOHlwrC43vXLjKFjPwc3hOgHvmm6a
I9MBi4s+SZpDbLyNWZlNJ84HvAvcCLwI0ppefLnUr1IiKuBx4FekTEmxHxrbJnqne+40FS1jySk5Q1Iycpa0ZOU
taMnKSsGTlJWTNyKlxE7BoRoyuX91vRJ6tExJoRcezH2MdPIuLEapcvsc1VEXHwv7Cvrn4KSNth5PSxVT4x5V+
SUrozpXTeCjZZE/iXIyctj5HTUipHKi9FxG8j4sWIuDkiVq2smxwR50fEROCQiBgYEY9GxMSIuCkiVq9st1flNiYC
B7W47aERcWnl8l8mci4raIeLryMwA4D9gsIv4UET+vbHdSRDwREc9ExFktbuu0iHgIh4GelRxv46s3M7TEXHL
wvtUsWdETKjc3j6V7Rsi4uct9v2dT/rfVq3PyGl5egD/m1LaAnifxY+uZqaUtgMeAE4H9qxcnwCcEBGrAJcD+w
J9gfWWs4//AR5MKbfUBtgOeB04BXk0pbZNSOikiBgLdaP4Iqm2AvhGxS0T0pfktb9sAewPbV3Gfbk0pbV/Z34
tAy3cLdK3s48vAiMp9+BYwO6W0feX2j4yITarYj+pIu7IHUN2aklIaV7l8LXAccGHl+g2Vf3eg+XZ3VCTvVRU
3LUE3g9pfRngIi4FjhqGfvYHfgGQEppATA7ItZaYpuBle+nKtdXpzl6nYDbUkp/q+yjmvf3f39o6In9J8Srw6zW+d

```
W+jGlFIT8OeIeK1yHwYCW7d4vO5TlX2/UsW+VCeMnJZnyff7tbw+t/JvAPenlIa03DAitqnhHAGcm1IaucQ+f
vAxbusq4ICU0tMRMRTYtcW6Zd3fAL6fUmoZQyKi68fYt0ri6aqWZ6OI2Kly+TDg4WVs8xiwc0RsDhARq0VEd
+AloGtEbFbZbsgyfhdgDHBM5XcbIuJTQCPNR2kL3Qsc0eKxvi4RsS4wFjggIjpGRCeaT40/SidgWkSsDBy+xLpD
ImKlysybAi9X9n1MZXsiontErFbFflRHjJyW52XguxHxIrAW8KslN0gpTQeGAtdHxDNUTlVTSv+g+fT0rsoTD+8
uZx/HA7tFxLPAk0CvlNJMmk9/n4uIn6eU7gOuAx6tbHcz0CmlNJHm0+angXto/oiqj3IG8DgwjuYQt/QXYHzlt
o6u3IdfAy8AEysvGRmJZz9tjp9CoqVUTsdGp5R6lz2L9El5JCcpax7JJGXNyEnKmpGTlDUjJylrRk5S1oycpax7JScqaR3KSsmbkJGXNyEnKmpGTlDUjJylrR
k5S1v4fqFlJ4TT5IccAAAAASUVORK5CYII=\n",
      "text/plain": [
       "<Figure size 360x360 with 1 Axes>"
      ]
     },
     "metadata": {
      "needs_background": "light"
     },
     "output_type": "display_data"
    }
   ],
   "source": [
    "from mlxtend.plotting import plot_confusion_matrix\n",
    "fig, ax = plot_confusion_matrix(conf_mat=cm ,  figsize=(5, 5))\n",
    "plt.show()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "id": "4b9976f9",
   "metadata": {
    "papermill": {
     "duration": 0.704178,
     "end_time": "2021-12-21T16:54:06.131840",
     "exception": false,
     "start_time": "2021-12-21T16:54:05.427662",
     "status": "completed"
    },
    "tags": []
   },
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.7.12"
  },
  "papermill": {
   "default_parameters": {},
   "duration": 180.885587,
   "end_time": "2021-12-21T16:54:09.492708",
   "environment_variables": {},
   "exception": null,
   "input_path": "__notebook__.ipynb",
   "output_path": "__notebook__.ipynb",
   "parameters": {},
   "start_time": "2021-12-21T16:51:08.607121",
   "version": "2.3.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 5
}
```