

```
{  
  "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))\n"
]
},
{
  "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": "W6My8YVOpFYI",
    "papermill": {
      "duration": 0.091685,
      "end_time": "2021-12-21T16:51:26.568839",
      "start_time": "2021-12-21T16:51:26.448369"
    }
  }
}
```

```
"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",
"   </tr>\n",
" </tbody>\n",
"</table>\n"
```

```
"<td>1</td>\n",
"<td>1</td>\n",
"<td>101348.88</td>\n",
"<td>1</td>\n",
"/>\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",
"/>\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",
"/>\n",
"<tr>\n",
"<th>3</th>\n",
"<td>4</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",
"    </tr>\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",
"/>\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",
"/>\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": "2021-12-21T16:51:26.8144458Z",
      "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')\"\n        ]
      },
      "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",
"/>\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",
"/>\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",
"/>\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
0	619	42	2	0.00	1	1
1	608	41	1	83807.86	1	0
2	502	42	8	159660.80	3	1
3	699	39	1	0.00	2	0
4	850	43	2	125510.82	1	1

\n,

	IsActiveMember	EstimatedSalary	Exited	Geography_France
0	1	101348.88	1	1
1	1	112542.58	0	0
2	0	113931.57	1	1
3	0	93826.63	0	1
4	1	79084.10	0	0

\n,

	Geography_Germany	Geography_Spain	Gender_Female	Gender_Male
0	0	0	1	0
1	0	1	1	0
2	0	0	1	0
3	0	0	1	0
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": {  
            "text/html": "

" i m a g e / p n g " :



iVBORw0KGgoAAAANSUhEUgAAAYgAAAEGCAYAAAB/+QKOAAAAOXRFWHRTb2Z0d2FyZQ  
BNYXRwbG90bGliIHZlcnNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACX  
BIWXMAAAsTAAALEwEAmpwYAAAWi0lEQVR4nO3df7RdZX3n8ffHBCIKKoSoFLDBgrapWsWAO  
rV2VUcEOyW14hKsFVs6OI50deq42qizKKWuruLMSGcqVmIlw2CnBG11MtN0okjVjnUgQQUMGIiAE  
mA0/BgQFULgO3/sHTkcnntzudydcy95v9Y63H2e/ex9vjv7cD93/zjPSVUhSdK4J026AEnS/GRASJKaDAh  
JUpMBIUlqMiAkSU2LJ13AXDnwwANr+fLlky5DkhaUK6644vaqWtaa94QJiOXLI7Nx48ZJlyFJC0qSb00  
1z1NMkqQmA0KS1GRASJKaDAhJUpMBIUlqMiAkSU0GhCSpyYCQJDUZEJKkJgNigs75V5dOugRJm  
pIBIUlqMiAkSU0GhCSpyYCQJDUZEJKkJgNCktRkQEiSmgwISVKTASFJajIgJEINgwZEkmOTbE6yJcn  
qxvxXJflKkh1JThibd3KS6/vHyUPWKUl6tMECIski4BzgOGAFcFKSFWPdvg28HfhvY8seAPw+8DLgaO  
D3k+w/VK2SpEcb8gjiaGBLVd1QVduBi4BVox2q6qaqugp4aGzZ1wGfraq7q+ou4LPAsQPWKKkaM2RA  
HAzcPPJ8a982Z8smOTXJxiQbt23bNutCJUmPtqAvUlfVeVW1sqpWLlu2bNLISNITypABCQtw6MjzQ/q2


```

oZeVJM2BIQNiA3BEksOS7A2cCKyd4bLrgWOS7N9fnD6mb5Mk7SaDBURV7QBOo/vFfi1wcVVtSnJm  
kuMBkhyVZCvwJuDcJJv6Ze8E/pAuZDYAZ/ZtkqTdZPGQK6+qdcC6sbbTR6Y30J0+ai17PnD+kPVJkqa2  
oC9SS5KGY0BIkpoMCElSkwEhSWoyICRJTQaEJKnJgJAkNRkQkqQmA0KS1GRASJKaDAhJUpMBIU  
lqMiAkSU0GhCSpyYCQJDUZEJKkJgNCktRkQEiSmgwISVKTASFJajIgJEINBoQkqcmAkCQ1GRCSpC  
YDQpLUZEBIkpoMCElSkwEhSWoyICRJTQaEJKlp0IBIcmySzUm2JFndmL8kyZp+/mVJlvfeyW5IMnV  
Sa5N8t4h65QkPdpgAZFkEXAOcBywAjgpyYqxbqcAd1XV4cDZwFl9+5uAJVX1QuClwDt2hockafcY8gji  
aGBLVd1QVduBi4BVY31WARf0058EXpMkQAFPTbIY2AfYDtwzYK2SpDFDBsTBwM0jz7f2bc0+Vb  
UDuBtYShcW3wduA74N/IequnP8BZKcmmRjko3btm2b+y2QpD3YfL1IfTTwIPBjwGHAv03y3PFOVXV  
eVa2sqpXLli3b3TVK0hPakAFxC3DoyPND+rZmn/500tOBO4C3AP+rqh6oqu8CXwJWDlirJGnMkAGxA  
TgiyWFJ9gZOBNaO9VkLnNxPnwBcWIVFd1rp1QBJngq8HPjGgLVKksYMFhD9NYXTgPXAtcDFVbUp  
yZIJju+7fQxYmmQL8G5g562w5wD7JtlEFzR/WVVXDWrJOnRFg+58qpaB6wbazt9ZPo+ultax5e7t9UU  
Sdp95utFaknShBkQkqQmA0KS1GRASJKaDAhJUpMBIUlqMiAkSU0GhCSpyYCQJDUZEJKkJgNCktRk  
QEiSmgwISVKTASFJajIgJEINBoQkqcmAkCQ1GRCSpCYDQpLUZEBIkpoMCElSkwEhSWoyICRJTT  
MKiCR/m+QXkxgokrSHmOkv/I8AbwGuT/LHSZ4/YE2SpHlgRgFRVZdU1a8CRwI3AZck+ackv55kryELI  
CRNxoxPGSVZCrwd+E3gq8B/oguMzw5SmSRpohbPpFOSTwHPBy4EfqmqbunrUmycajiJEmTM6OAA  
P68qtaNNiRZUIX3V9XKAeqSJE3YTE8xfaDR9uVdLZTk2CSbk2xJsroxf0mSNf38y5IsH5n3oiRfTrIpydV  
JnjzDWiVJc2DaI4gkzwYOBvZJ8hIg/aynAU/ZxbKLgHOA1wJbgQ1J1lbVNSPdTgHuqqrDk5wInAW8Ocl  
i4OPAr1XVlf31jwce++ZJkmZrV6eYXkd3YfoQ4EMj7d8D3reLZY8GtlTVDQBJLgJWAaMBsQo4o5/+JP  
DhJAGOAa6qqisBquqOXW2IJGluTRsQVXUBcEGSN1bV3zzGdR8M3DzyfCvwsqn6VNWOJHcDS4Hn  
AZVkBAMuKiQpj+AklOBu4FeM5znvMYy5MkTWdXp5jeWIUfB5Yneff4/Kr6UGOxuarrlcBRwA+Azy  
W5oqo+N/b65wHnAaxcubIGqkWS9ki7ukj91P7nvsB+jcd0bgEOHXI+SN/W7NNfd3g6cAfd0cYXq+r2qvo  
BsI7uMxeSpN1kV6eYzu1//sEs1r0BOCLJYXRBClDcB2j1gIn090RdQJwaVXtPLX0u0meAmwHfh44ex  
Y1SJJmaaaD9X0wydOS7JXkc0m2JXnrdMtU1Q7gNGA9cC1wcVVtSnJmkuP7bh8DlibZArwbWN0vexfd  
RfENwNeAr1TV381i+yRJsTTD8odU1W/m+QNdGMx/QrwRpbUafUf7hu3Vjb6SPT9wFvmmLZj+9q/Z  
Kk4cz0g3I7g+QXgU9U1d0D1SNJmidmegTxP5N8A/gh8M4ky4D7hitLkjRpMx3uezXwz4CVVfUA8H26  
D7lJkp6gZnoEAfCTdJ+HGF3mv85xPZKkeWKmw31fCPwE3R1FD/bNhQEhSU9YMz2CWAmsoCo/rSxJ  
e4iZ3sX0deDZQxYiSZpfZnoEcSBwTZLLgft3NlbV8VMvIklayGYaEGcMWYQkaf6ZUUBU1ReS/DhwR  
FVd0o+RtGjY0iRJkzTTsZj+Jd0X+pzbNx0MfHqgmiRJ88BML1K/C/hZ4B6AqroeeOZQRUMSJm+mAXF  
/VW3f+aT/sJy3vErSE9hMA+ILSd4H7JPktcAngP8xFmSpEmbaUCsBrYBVwPvoBvC+98NVZQkafJmeh  
fTQ0k+DXy6qrYNW5IkaT6Y9ggmTOS3A5sBjb33yZ3+nTLSZIWvl2dYvoduruXjqqqA6rqAOBlwM8m+  
Z3Bq5MkTcyuAuLXgJOq6sadDVV1A/BW4G1DFiZJmqxdBcReVXX7eGN/HWKvYUqSJM0HuwqI7bO  
cp11x5HRJ89yu7mL6mST3NNoDPHmAevYc//ts4MWTrkKSpjRtQFSVA/IN5fbrMCAkzWcz/aCcJGkPY  
0BIkpoMCElSkwEhSWoyICbF21wlzXMGhCSpyYCQJDUNGhBJjk2yOcmWJKsb85ckWdPPvyzJ8rH5z0  
lyb5L3DFmnJOnRBguIJUAc4DjgBXASUIWjHU7Bbirqg4HzgbOGpv/IeDvh6pRkjS1IY8gjga2VNUN/fdZ  
XwSsGuuzCrign/4k8JokAUjyy8CNwKYBa5QkTWHIgDgYuHnk+da+rdmnqnYAdwNLk+wL/B7wB9O9  
QJJTk2xMsnHbNr/oTpLm0ny9SH0GcHZV3Ttdp6o6r6pWVtXKZcuW7Z7KJGkPMaPvpJ6IW4BDR54f0r  
e1+mxNshh4OnAH3bfWnZDkg8AzgIeS3FdVHx6wXknSiCEDYgNwRJLD6ILgROAtY33WAicDXwZO  
AC6tqgJ+bmeHJGcA9xoOkrR7DRYQVbUjyWnAemARcH5VbUpyJrCxqtYCHwMuTLIFuJMuRCRJ88C  
QRxBU1Tpg3Vjb6SPT9wFv2sU6zhikOEnStObrReonvu5uXkmatwwISVKTASFJajIgJEINBoQkqcmAkC  
Q1GRCSpCYDQpLUZEBIkpoMCElSkwEhSWoyICRJTQaEJKnJgJiUqkIXIEntMiAkSU0GhCSpyYCQJ  
DUZEJKkJgNCktRkQEiSmgwISVKTASFJajIgJEINBoQkqcmAmJRk0hVI0rQMCElSkwEhSWoyICRJTQ  
aEJKlp0IBIcmySzUm2JFndmL8kyZp+/mVJlvftr01yRZKr+5+vHrJOSdKjDRYQSRYB5wDHASuAk5KsG

Ot2CnBXVR0OnA2c1bffDvxSVb0QOBm4cKg6JUltQx5BHA1sqaobqmo7cBGwaqzPKuCCFvqTwGuSpK  
q+WIW39u2bgH2SLBmwVknSmCED4mDg5pHnW/u2Zp+q2gHcDSwd6/NG4CtVdf/4CyQ5NcnGJBu3bd  
s2Z4VLkub5ReokP0132ukdrfIVdV5VrayqlcuWLdu9xUnSE9yQAXELcOjI80P6tmafJIuBpwN39M8PAT4  
FvK2qvjlgnZKkhiEDYgNwRJLDkuwNnAisHeuzlu4iNMAJwKVVVUmeAfwdxLqqvjRgjZKkKQwWEP0  
1hdOA9cC1wMVVtSnJmUmO77t9DFiaZAwwbmDnrbCnAYcDpyf5Wv945lC1SpIebfGQK6+qdcC6sbbT  
R6bvA97UWO4DwAeGrE2SNL15fZFakjQ5BoQkqcmAkCQ1GRCSpCYDQpLUZEBIkpoMCEISkwEhS  
WoyICRJTQaEJKnJgJAKNRkQkqQmA0KS1GRASJKaDAhJUpMBIUlqMiAmJpMuQJKmZUBIkpoMCE  
ISkwEhSWoyICYIXoOQNL8ZEJKkJgNCktRkQEiSmgwISVKTATFhm269e9IIISFKTATFhF11+86RLkK  
QmA2JivM1V0vxmQEiSmgyISekPIJ7kgYSkeWrQgEhybJLNSbYkWd2YvyTJmn7+ZUmWj8x7b9++Ocn  
rhqxzMlpk2P5gTbgOSWobLCCSLALOAY4DVgAnJVkx1u0U4K6qOhw4GzirX3YFcCLw08CxwEf69T3  
hbL3rB5MuQZKaFg+47qOBLVV1A0CSi4BVwDUjfVYBZ/TTnwQ+nCR9+0VVdT9wY5It/fq+PEShP9z  
xQz5z02fY/uD97LvxOva643sUM/jLvh7uUyP/fXh2PartR7ZeBfwKL/7Geaw/e02jQx4xFNGL2r3q3wsz  
llji6Kz+jfZYLqMZc3v7en89j23fwZ5muKQqYr8DHXPruN/d5Rz2PH0qfNatn56AVLX8BPLf2pOV/vkAF  
xMDB6D+dW4GVT9amqHUnuBpb27f9nbNmDx18gyanAqf3Te5Nsak2hT3rKk5629zP3PmLRQ7D8O7N  
Zw+zcudJ/NgD9+y+FxzAnQ/u4IBFQ76Nhuc2zA+7cxvu3A/u2nfu1/vg9x5k0X67/2THQ/c/9L3tt22/bpaL//  
hUMxb0O6qqzgPOm3Qds5Vk451VKyddx+ORZOOtDzzgNkyY2zA/JNm4/fbtC3obRg15kfoW4NCR54f0  
bc0+SRYDTwfumOGykqQBDRkQG4AjhyWZG+6i85rx/qsBU7up08ALq3uxP1a4MT+LqfDgCOAywes  
VZI0ZrBTTP01hdOA9cAi4Pyq2pTkTGBjVa0FPgZc2F+EvpMuROj7XUx3QXsH8K6qenCoWidowZ4eG  
+E2zA9uw/zwRNiGH0k99ts+JEI7AD9JLULqMiAkSU0GxATsagiSSUhyU5Krk3wtyc+7YAkn01yff9z/74  
9Sf5zX/9VSY4cWc/Jff/rk5w80v7Sfv1b+mUf98e5kpyf5LtJvj7SNnjNU73GHG7DGULu6ffF15K8fmRecwi  
aqd5T/U0il/Xta/obRqYd5mYW23Bokn9Ick2STUl+u29fMPtimm1YUPtizIWVj934oLtg/03gucDewJXAinl  
Q103AgWNtHwRW99OrgbP66dcDf0/3MdaXA5f17QcAN/Q/9++n9+/nXd73Tb/scXNQ86uAI4Gv786ap3  
qNOdyGM4D3NPqu6N8vS4DD+vfRouneU8DFwIn99EeBd/bT/xr4aD99IrDmcWzDQcCR/fR+wHV9rQtm  
X0yzDQtqX8z1Y+IF7GkP4BXA+pHn7wXeOw/quolHB8Rm4KB++iBgc99LnDSeD/gJODckfZz+7aDgG  
+MtD+i3+Osezmp/OU6eM1TvcYcbsNUv5Qe8V6hu0PwFVO9p/pfprcDi8ffezuX7acX9/0yR/vkvwOvXYj  
7orENC3pfPN6Hp5h2v9YQJI8aRmQCCvhMkivSDWEC8Kyquq2f/r/As/rpqbZhuvatjfYh7I6ap3qNuXRaf/r  
1/JHTJo91G5YC/6+qdjS24RHD3AA7h7l5XPrTIy8BLmOB7ouxbyAFui/mggGhnV5ZVUfSjb77riSvGp1Z  
3Z83C+qe6N1R80Cv8WfATwAvBm4D/uMcr38QSfYF/gb4N1X1iEHGFsq+aGzDgtwXc8WA2P3m5TAi  
VXVL//O7wKfoRs/9TpKDAPqf3+27T7UN07Uf0mgfwu6oearXmBNV9Z2qerCqHgL+nG5fzGYb7gCek  
W4Ym/FtmGqYm1JshfdL9a/qqq/7ZsX1L5obcNC3BdzyYDY/WYyBMluleSpSfbbOQ0cA3ydRw6FcjLde  
Vn69rf1d6O8HLi7P8xfDxyTZP/+UPwYuvOstwH3JH15f/fJ20bWNdd2R81Tvcac2PkLr/cGun2x83VbQ9A  
031P9X9T/QDeMzXitUw1zM5t6QzcqwrVV9aGRWQtmX0y1DQttX8y5SV8E2RMfdHdxXEEd3t8P750E9z  
6W72+JKYNPOmujOg34OuB64BDigbw/dl0F9E7gaWDmyrt8AtvSPXx9pX0n3P9c3gQ8zBxfhgL+mO+x/  
gO6c7im7o+apXmMOt+HCvsar6H55HDTs//19PZsZuRNsqvduv28v77ftE8CSvv3J/fMt/fznPo5teCXdqZ2r  
gK/1j9cvpH0xzTYsqH0x1w+H2pAkNXmKSZLUZEBIkpoMCEISkwEhSWoyICRJTQaE9khJnp3koiTf7Ic  
XWZfkebNc139JckI//RdJVvTT7xvr9/5+pNCr+pFBX/b4t0QazmBfOSrNV/2Hoj4FXFBVJ/ZtP0M3js91/fP  
F9fC4OTNWVb858vR9wB/163sF8C/oRgy9P8mBdKN9Pp7tmFWN0kx5BKE90S8AD1TVR3c2VNWVw  
KIk/5hkLXBNkkVJ/n2SDf1f/e+AH32fwYfTjfl/CfDMnetJ8vkkK5P8MbBPf6TwV3Qjjd5eVff3r3d7Vd3aL  
3NUkn9KcmWSy5Psl+TJSf4y3XcfgDXJL/R9355kbZJLgc/1n4I/v1/uq0lW7Z5/Qu0JPILQnugFwBVTzDsS  
eEFV3ZhuVNu7q+qoJEuALyX5DN1In8+n+06AZwHXAOePrqSqVic5rapeDD8aBO70JNfRfeJ3TVV9oR  
+OYQ3w5qrakORpwA+B3+5WUy9M8pN0I+3uPAV2JPCiqrozyR/RDc3wG0meAVye5JKq+v5c/ENpz2Z  
ASI90eVXd2E8fA7xo5/UFukHUjqD7kp+/rqpHgVv7v+anVVX3Jnk8HN0RzBr0n3b2BXAbVW1oe93D0

CSVwJ/2rd9I8m3gJ0B8dmqunOkxuOTvKd//mTgOcC1s9t86WEGrPZEm3h40LRxo395B/itqlo/2iEjXzv5  
WPSB8nng80muphugbaojmemM1/jGqto8m5qk6XgNQnuiS4ElefiLkUjyIrq/7ketB96Zbhokjwv3Wi3XwT  
e3F+jOIjuiKDlgZFln5/kiJF5Lwa+Rf+NaEmO6vvtl27I538EfnXn69IdFbRCYD3wW/2Fd5K8ZIb/BtIueQSh  
PU5VVZI3AH+S5PeA++i+cvXTY13/gu7rQL/S/wLeBvwy3R1Qr6a79vBt4MtTvNR5wFVJvgJ8CPjT/jrB  
DrqRO0+tqu1J3tzP24fu+sM/Bz4C/FI/pLEDeHt/99P4a/wh8Cf96zwJuJHubinpcXM0V0lSk6eYJEINBoQkq  
cmAkCQ1GRCSpCYDQpLUZEBIkpoMCEIS0/8HJ0txqYVxJvwAAAAASUVORK5CYII=\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))\n"
]
},
{
"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": "image\_content\_base64",  
"text": "iVBORw0KGgoAAAANSUhEUgAAAYIAAAEGCAYAAABo25JHAAAAOXRFWHRTb2Z0d2FyZQB  
NYXRwbG90bGliIHZlcnNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXB  
IWXMAAAAsTAAALEwEAmpwYAABHq0IEQVR4nO3deXydZ3ng/d/1nH3RvlvyIlm2Y8d2NmcnJBACo  
VCWwjsspS1lCu106Ke8Zaaly0sppZ22hLbTQjtQhoHSThMgQFNlcBKyOSROvO+bbMva9+3s6/3+cSRbt  
o6ko+XoyNL1/XxkS+e5n+e5dCSd69y7GGNQSim1elmFDkAppVRhaSJQSqlVThOBukqtcpoIlFJqlNEoJ  
RSq5y90AHMVWVlpdmwYUOhw1BKqevKgQMHBowxVdmOXxJYMOGDezf7/QYSil1HVFR5Nd  
0ybhpRSapXTRKCUUqucJgKlIfrlNBEopdQqp4IAKaVWOU0ESim1ymkiUEqpVU4TgVJKrXKaCFROek  
I9hQ5BKZUnmgjUrDoCHbz1e2+lK9hV6FCUUnmgUDN6oX2FwB46uJTBY1DKZUfmjUrJ5te5YH1z3  
I7tbdhQ5FKZUHmgjUrM6PnOe+hvvoCHYUOhSIVB5oIlAziqfiBONB6v31xJlxwolwoUNSSi0yTQRqRn  
3hPsrd5VhiUeGpoCeso4eUWmk0EagZ9YZ7KXeXA1DuLtdhpEqTQJoI1Ix6Q72UukuBTCLoDfUWNiCl1  
KLTRKBm1BvupdRVCKCJq4TuUHdhA1JKLTpNBGpG3aHuy4mg3F2uk8qUWoE0EagZ9YR6LjcNlbhK  
GloOFTYgpdSi00SgZjQSG6HIUQSAz+FjJDZS2ICUUotOE4Ga0VhsDJ/DB2QSwWhstMARKaUWmyY  
CNaOx+JVE4Hf4GYuPFTgipdRi00SgZhSIB66qEQTiAYwxBY5KKbWYNBGoacVSMZLpJC6bCwC7Zc  
dpcxJO6jITSq0kmjUtMZiY/idfkTk8mN+h1/7CZRaYTQRqGmNxkYvNwtN8Dv8OnJiqRVGE4Ga1lh8D  
L/Df9VjOnJiqZVHE4GaVrYagdfhZTSuiUCplUQTgZrWaHwUr8N71WN+h5+xmA4hVWol0USgpjUaG8  
VrvzoReOwebRpSaoXJayIQkYdF5IyItIjIZ7IcXyciz4vIIRE5KiI/l8941NxkSwRuu5tAPFCgiJRS+ZC3RCAi  
NuArwNuBbcCHRGtbNcX+CPiOMeYW4IPAP+QrHjV3gXgAj8Nz1WMeu4dgIligiJRS+ZDPGsEdQIsx5  
oIxJg48Crz7mjIGKB7/vATQNY6XkVAihNvmuo9j2jNQKIVph8JoJ6oH3S1x3jj032OeAjItIBPAn8VrY  
LicgnRGS/iOzv7+/PR6wqi0AigMd+dY3AbXdrjUCpFabQncUfAr5pjGkAfg74tohMickY8zVjzC5jzK6qqq  
oID3K1CsVDuO1TawShRKhAESml8iGfiaATWDvp64bxxyb7z8B3AIwxrwJuoDKPMak5CCVDU2oEHru  
HYFxrBEqtJPlMBPuATSLSKCJOMp3BT1xTpg14EEBEtpJBNr2s0yEElNrBG6bm1BSawRKrSR5SwTG  
mCTwSWA3cIrM6KATIvJ5EXnXeLFPAx8XkSPAvwEfNbrG8bIRSoTw2KbWCMIJXX1UqZXEns+LG2  
OeJNMJPPmxz076/CRwbz5jUPMXToS1j0CpVaDQncVqmUqbNNFkdEoisFt2jDHEU/ECRaaUWmyaCF  
RWkWQE192Fdc0gLhHB6/BqrUCpFUQTgcoqGA9OGTE0QWcXK7WyaCJQWWUbOjpB+wmUWlk0Ea  
isQvGZE4HOJVBq5dBEoLIKJqZvGtJIJpRaWTQRqKyyLTg3wWVz6VwCpVYQTQQqq3AyjMvuynrMZ  
XMRTmoiUGql0ESgsgonwrhsMyQCrREotWJoIlBZhZNhnJYz6zGnzak1AqVWEE0EKqvZagQ6fFSplUM  
TgcoqmAhO20fgtrk1ESi1gmgiUFnNWCOwa2exUiuJJgKVVTgRnnb4qNYIIfpZNBGorGYbPhpJRJY4Iq  
VUvngiUFmFk9o0pNRqoYIAZTVTH4Hb5iaS1BqBUiuFJgKVVSQZmbIpzQSdWazUyqKJQGUVSUamr  
xHY3dpHoNQKoolAZRVOTj9qyGVzadOQUiuIJgKVVTQZnbZG4LQ5iafipE16iaNSSuWDJgI1RSKVIG  
VS2C171uOWWDhtTqLJ6BJHppTKB00EaopwMozH7kFEpi3jtru1w1ipFUITgZoikoxM2z8wwW3X2cVK  
rRSaCNQU4WR42qGjE7TDWKmVQxOBmiKSmH7o6ARNBEqtHJoI1BQzLS8xQdcbUmrl0ESgpphpMt  
kErREotXJoIlBTRJIRnLbs21RO0O0qlVo5NBGoKXJNBFojUGpl0ESgpsgpEViaCJRaKTQRqCkiyQhOS2  
sESq0WmgjUFLnUCFw2F+GE9hEotRJoIlBTzLQpzQTtLFZq5dBEoKbIJRHo8FGIVg5NBGqKcDKsTUN  
KrSJ5TQQi8rCInBGRFhH5zDRI/pOInBSREyLyf/MZj8pNrjOLtWIIqZUh+4Lzi0BEbMBXgIeADmCfiDx  
hjDk5qcwm4PeBe40xwyJSna94VO50rSGIVpd81gjuAFqMMReMMXHgUeDd15T5OPAVY8wwgDGmL  
4/xqBzlMSErjWk1IqQz0RQD7RP+rpj/LHJNgObReRnIrJXR7OdiER+YSI7BeR/f39/XkKV03Iafio3UU  
kpYIAqZWg0J3FdmAT8ADwIeCfRKT02kLGmK8ZY3YZY3ZVVVUtbYSrUDQ1/X7FE1w2l25VqdQKk  
c9E0AmsnfR1w/hjk3UATxhjEsaYi8BZMolBFVAuTUM6s1iplSOfiWAfsElEGkXECXwQeOKaMj8kUxt  
ARCrJNBVdyGNMKgfRpNYIIfpN8pYIjDFJ4JPAbuAU8B1jzAkR+byIvGu82G5gUEROAs8D/90YM5iv"}

Scanned with CamScanner

mFRuckkETstJIp0glU4tUVRKqXzJ2/BRAGPMk8CT1zz22UmfG+B3xj/UMpBIJ0ibNHZr5l8NEcnUCIJRf  
JZviaJTSuVDoTuL1TITUZx2V2IyKxldS6BUiuDJgJ1Iugygtvmzqmsy66JQKmVQBOBukokGcFln71/YI  
LWCJRaGTQRqKvkMnR0giYCpVYGTQTqKpoIIfp9NBGoq+Sy4NwEp82p6w0ptQLklAhE5Psi8g4R0cS  
xwuWyztAErREotTLk+sL+D8CHgXMi8hcisiWPMakCiqRm37h+gtNyEk3p7GKlrnc5JQJjzLPGmF8EbgV  
agWdF5BUR+VURceQzQLW05tJH4LTrekNKrQQ5N/WISAXwUeDXgEPA/ySTGJ7JS2SqICKJ3JuGnJY  
mAqVWgpyWmBCRHwBbgG8DP2+M6R4/9JiI7M9XcGrpRZIRHLbcKn1Om1P3LVZqBch1raF/GI836DI  
RcRljYsaYXXmISxVIOBnOvY9AE4FSK0KuTUNfyPLYq4sZiFoewonZN66foKOGLFoZZqwRiEgtme0IP  
SJyCzCxElkx4M1zbKoAIskIpa7SnMq6bC76kyt769Czw2f58qEv0zLSwtbyrXzq1k+xtnjt7CcqdR2ZrUbwN  
uARMruL/TXwpfGP3wH+IL+hqULQmcVXvNr1Kh/7yceo9dXy8R0fp9hZzIef/DAnB08WOjSIfWMNQJ  
jzLeAb4nI+4wxjy9RTKqAdNG5jLaxNn73pd/IN276DbaUZ6bNNBQ1UO2t5ree+y2+/67vU+IqKXCUSi2  
OGWsElvKR8U83iMjvXPuxBPGpJRZOhnMfPrpC9y02xvBHP/sjHt7w8OUkMGFX7S52VO7gS/u/VKDo  
lFp8szUNTWw95QeKsnyoFSaSjOCyVneN4JILzzASHeEt69+S9fh7m9/L8+3Pc2743BJHplR+zNY09NXx/  
/9kacJRhTaXtYZWYo0gbdL83aG/432b3oc1zdJaXoeXh9Y/xFePfpVH7n9kiSNUavHluujcX4lIsYg4ROSnI  
tI/qdlIrSC5bFw/wW1zE02urLWGXMh/AbvYubHixhnL3d9wP690vkJXsGtpAlMqj3KdR/BWY8wY8E4ya  
w01A/89X0Gpwl2ncXpNHQfgeOPw/HvQ/dRMczvt/vnk//Mg+sfnHXPZq/Dy91r7uZ7Z7+Xt1iUWiq5ziy  
eKPcO4LvGmNFcNjdX15+51Aictszqo8aYnDa7n5PICOz9R9j/DXB4oHR9ZhbL4AWwbPDQn8C298Ai3r  
dtrI2W4RY+sfMTOZW/r+E+/vbA3/KbN/8mdivXPyWllp9cf3t/JCKngQjwX0SkClhZbQIKYwyxVCznRG  
C37FhYJNKJnPsVcnLyCfjxp6HuJnjL56B03eQgofcYPPsncOYpePdXIMe1kWbz/XPf5641d+Gwcrtevb+e  
MncZe7v38ob6NyxDKEoVQq7LUH8GuAfYZYxJACHg3fkMTC29aCqKw3JM20majcvuWrz1hJJJeOoz8J  
PPwBv/O9z721cnAcjUAGp3wtu/CMOX4NFfhHRqwbc2xvDkxSe5q+6uOZ13e+3t/Mf5/1jw/ZUqpLnsOH  
YD8AER+WXg/cBb8xOSKpRIMoLb7p7TOW6be3H6CVIJ+N5HoX0v/NwjUL115vIONzzwGQj1wdN/tO  
DbHx04iiUW64rWzV54kjtq7+DFjheJpWILjkGpQsl11NC3ySw18Qbg9vEPXXV0hZnL8hIT3PZFSATpN  
HzvYxDsgzd/Flw5TIGx7PDG34UTP8w0Ey3AM63PcGv1rXPu6yhxlCuaB17u/Yu6P5KFVKufQS7gG3G  
5HG4hiq4uaw8OmFRRg498//BcCu85U/m3t7vKoJ7Pgn/8duw7i7wlM359sYYnm17ll/b8WtzPhfg5uqb2d26  
m/vX3j+v85UqtFyaho4DtfkMRBXefGoELpuLcHIBfQTHvgcnfwgP/P78O31rd0LDHfDTz8/r9AujF4ilYn  
NuFppwS/Ut7OncQ2oR+iqUKoRcE0ElcFJEfovIEExMf+QxMLb25zCGYsKDZxYPn4cn/lmneybU5aDo3f  
QhO/AD6z8z51Jc6XmJH5Y55D4Gt9FRS7Czm+ODxeZ2vVKHl2jT0uXwGoZaHJa0RpNPwg1+HHf8Jyp  
vmfv613MWw/X2ZjuNf/O6cTn2x40XuWXPPgm6/vXI7L7a/yE1VNy3oOkoVQq7DR18kM6PYMf75PuB  
gHuNSBTdfRBBJzKNG8PrXIBGBG94x93Ons+XnoOtQZiZyjsKJMCcHT7K1fJZRSrPYUbmDlzpeWtA1  
ICqUXEcNfRz4HvDV8YfqgR/mKSZVIJFkJOf9iifMq2ko0Asv/A+4+7/CHOYszMrmzMw2fuEvcj7lQO8B  
Gosb5zxs9lobSzfSEexgMDK4oOsoVQi5/hX+V+BeYAzAGHMOqM5XUKowIskITvscE4E1j0Tw7Oeg+U  
EoycOWj5vfBm17c+4reLXr1Sl7DsyH3bKztXwrr3brVt7q+pNrIogZY+ITX4iIHdChpCvMfGsEc5pZ3HM  
Mzu2GHR+YY3Q5srth88Ow9x9yKv5K1ytsrVhYs9CELeVbeLVLE4G6/uSaCF4UkT8gs4n9Q8B3AZ1Xv  
8KEEqE5rxk0587iZ/8EdrwfnN45RjcHW96eWa00PDRjsZHoCF2hLjYUb1iU226r2Mbe7r3odBt1vck1EX  
wG6AeOAb8OPAkSF6/WlbnNaFsLmsNtb+eqRFsenge0c2BpwzW3gUH/3nGYgd6D7C5bPOirRxa660lZ  
VK0jrUuyvWUWiq5jhpKk+kc/k1jzPuNMf+ks4xXnnByfjOLc64RPP/nmdrAIq0WOqPNb4P9/zszTHUar/  
W8RnNp86LdUkTYWr6VfT37Fu2aSi2F2TavFxH5nIgMAGeAM+O7k302l4uLyMMickZEwkTkMzOUe5  
+IGBHR9YsKKJKY5/DRXDqLuw5D7wnY+OD8gpuryi2ZUUQXX5y2yP6e/WwpW3hH8WSbyzZrP4G6  
7sxWI/h/yWwWut0YU26MKQfuBO4Vkf93phNFxAZ8BXg7sA34kIhsy1KuCPht4LV5xK8W0XxqBG6bO  
7emoZcegW3vWpraAGSWq970Vtj3T1kPB+IB2gPtBcjZsKi3vaH8Bg70HtB+AnVdmS0R/BLwIWPMxYk  
HjDEXgI8AvzzLuXcALcaYC+Mjjh4l+x4Gfwr8JbrRTcGFk+E5LzHhsufQNDTcCq17YNPb5h/cfDQ+ABd  
egtDAIENH+o/QVNK06DuLVXoqcdqcXBisKjXVSqfZksEDmPMIL8iY0w/MNtbu3qgfdLXHeOPXSYit

wJrjTE/nulCIvIJEdkvIvv7+/tnua2ar3AijNs29/0IZk0Ee/8Rmt+S2XJyKTm9sO5OOPrYlEMHew/SVLoIS1tksalsE/t79ufl2krlw2yJID7PY7MSEQv4a+DTs5U1xnzNGLPLGLOrqqpqIbdVM5j3EhMz9RFEx+Dw/80s/1AIG98MB781ZcP7A70H2FS6KS+33FS2idd6tKVTXT9mSwQ3ichYlo8AsGOWczuByVNHG8Yfm1AEbAdeEJFW4C7gCe0wLpz57FDmss+y1tDRxzJ7D/sLNBG9ZjvEAtB9+PJDiXSCU0On2Fi6MS+33FK2hYO9B7WfQF03ZkwExhibMaY4y0eRMWa2pqF9wCYRaRQRJ/BB4PLS1caYUWNMpTFmgzFmA7AXeJcxRuvUBRJOzq9paNoagTGZxeU2v30RopsnsTJ9BYf/7+WHzgydocpThdeRn0ltVZ4qDIaOQEderq/UYlvcnrJJjDFJEfkksBuwAd8wxpwQkc8D+40xup/BMhNNRufcWeywHCTTSZLp5NSO10uvQDIGtbNVHucvkkhxrGOEUz0BukaiBKMJ0sbgtNvwOW2U+Zw0urbxxlOPEL7vjk8nO473DeagOQmU+wuWwzB/oOsLY4D+spKbXI8pYIAIwxT5KZhTz5saxzEIwxD+QzFjWzRCpB2qSxy9x+JUTk8r7FRc5rNpfZ9/XMEM55bvgyk+FwnB8c6mRPywD1JR7WVXjZUuPH67QjAsm0IRJPEYwlOT7moyIVzl9/8W/oqHkzjrqXuXPt5kWPabKNpRvZ37Of9zS/J6/3UWox5DURqOtHOBnGbXfPa5cutz0zl+CqRBAZhnPPwHu/Ov2J85A2hp8c7+Hxg53saCjhE/c1UeyefW6Cu/hNfC5whMeaPsQ/XTrB+UsbsN3ez70bK+e9M9lMNpVu4hvHv7Ho11UqHxZxMXh1PYskI3js8xvemXUI6dHvQMOuzM5hi2QkkuALPz7FC2f6+eW71/PQ1pqckgDAaM2dlPW8QnP5MC5Hkvd38b3DnTwyNNnCcaaSixbjhIaiBgajgwxFZ174TqnIQBOBAuY3h2CC254lERz81qIuJ3GmN8BnHj9Khc/JL929nkr/3Poy0g4/wYqd9F38N9b4GllX7uVX72nEZsEf/uAYvaOLO5/REovm0mYO9R1a1OsqlQ+aCBQwv1nFE1y2a1Yg7Tmemc1bt3NRYnvZC9f3H2Gt22v5YEt1VjzbMoZqbuHSwMvUetZB4DdEt62rZZb1pXy2SdO0Dk8j72XZ9BU0sTBXt3RVS1/mggUsPAawVVDSA//KzQ9sOBtKOOpNF998QL/cbSLX757PZuri2Y/aQbBips4QYgG6+rr3L6+nAe2VPH5H51a1JpBc2kzB3oPLNr1lMoXTQQKmN+CcxOuqhGknDsu9D0pgXF0x+M8cdPHKc3EOGj9zRS4ZtfbJMIBc45HGwdujj12M76Eu5truALT54kEEks+F6QqRG0jLQQ8UW5XpK5YsmAgXMb1OaCU6b80ofwYUXwFcJJQ3zjuVQ2wh/8INjNFf5+YVbGnDZF+fXtDXWTYW9mJqeV8i20+pt68rYVF3El545Qyo1/T4GuXLZXTT4GzgxcGLB11IqnzQRKGARawRHH4UNb5zXdVLG8Ni+dv7Xiy38ws313NW0uEM7z0YvscZVj7FseIezb27/pi3VJNKGx/a3Zz0+VxtLN2qHsVr2NBEoIFMjmOt+xROcNiehRAjiYTi7Gza8Ye73jyf5q5+c4VD7MB97QyPrKnzzimUmZyOXWOOsIIg9LO57OWsQTefVM9L57t53jn6ILv2VTapP0EatnTRKCATI1gvonAbXMTTATH7FNQuTmzZ/AcDEfi/PETJ7Fb8OE71+F35WfzmrPRNhqc1QTLt1Pcvx9JZe8Y9jltvGNHHV95voVQfGFzDJpLmznaf1QXoFPLmiYCBUAoEVrQqKFQIpSZRDbH2sBwOM7nnjjBxiofb99eh22BI42mM5QcI2ZilNtKSDn8RP3rKOmdfqnojVV+mqp8/MurlxZ033J3OU6bk0tjC7uOUvmkiUABEIwHFzSzOBAZhtaXYd1dc7hnki/8+BTbaku4b1NVXpZ6mHAueokGZ83lewQrdllW8dyM57z5hmoOtg1zqntsQfdulm3mcP/hBV1DqXzSRKAACCQCc96LYILH7iE00gq128Hpz+mcVNrw10+fpb7Uw73NFfO671xM9A9MCJc04wz34gx1TXuO227joa01fG3PBZILGEXUWNKoE8vUsqaJQAHjTUPzTARuu5tQoAPW3ZvzOY/uayOWTPPQtpq81gQmnI62Uu+YtDmO2AhWbJ+1VnBDXTE+p42fHO+e972bS5s53Hd43ucrlW+aCBQw9z6CRNhw7rEI+/8sQOzxKpIBJ6y9M6dzj3SM8NK5Ad5905p5LxcxF/F0gvZ479WJAAhW3Exp9x4kPf0EMgEe2lrLDw93MRye3+6sDUUN9IR7GI0tfBSSUvmgiUABmUSQax9BMmI49X/CiBiafsGN093PLWc/TSg++/nBWJJ/fOE8P79zDT7X0qyCfjHWRZW9DKd19WikhLucuLea4hk6jQFKh528M76WJ3/UO+von+hgmq6XYwydTGDSmbJ2y05TSRPHBo4t7BtRKK80EShgatNQIIP+/EI2mUkZzn0ngqdSqL7did0tlHr20F16lqd+kCSd5RyArvY0P/lhkm/+c4AtRRU0Vi7ePAGThv42i86zNuJZRoSejV6i3pl9z+RA5a2Utz897bXjByC+R6j3uqk+6+XIo8GszwvAwLE4J74eItSZouP5OOcejZBOZso2ljTqxDK1bOnGNArITCjz2D2kE5kmn8ClFDA3sP5hF+U3Zt5JG2NofTJKOm6ofsP4nINkFOdIG2erRrl7+A5efTHFvW+6+tfq0GspDr2epmxdnMFYlNoLaxhcE6eiPksHrDEQDBNuH2Ow3YYrOUaldGCZFNht4HaD3wfFfij2k/IWcXxvMbGI4PIaWg872PFgjOKKKy/WpyIXWe+qy/59l2yivP1pXIE2YkXrrjqWPA/JM4LzboO4hFBxlPRJw8Uf2Wn8+as38Rk+naDtqThrH3LhLrMwaUPnC3Fan4zS9C4PzaXN7OncM58fjVJ5p4IAAVc2rm9/NkY6CZs+6CYykKbt6RgDRxKU3+hg6HiC2Kih4UEnYo2/CPadxlFUS8IEuelO4WfPGErLutx4s434wCgvPxmirdfL1the+n/SxRsdQsJdxaln38jNYz+kyAxBOg3RGASCMDJGZ/XdXFj/DkrjHUQ

ddbRLDTvjT+NMhSCRhEgUwhFMKMKJtR9G6GHT0MuIx81I6UaO/fgWdvlewFXmwJQUYQLn2VK3  
HodnDGwW6ZRFfMiPJQlcvlGiZhu1+79P75YPYux2Ui4XiZSL2M9cOG83yPjKGxtq/fy0v5eScw7szwpr3  
+JCRBg5l+TCv0dpeHMmCQCIJdS9wcmlJ2MMHE2wcetGvnr0q9n3dlaqwPQ3UpFMJ0mkEhCyM3A4T  
OO73IgleKttbPh5i5GzKQYOJ3CVC9V3OLDskzp4e45ilW7AETyFsSe47YYxXnvGw9F/7yNqPDjtKbY7  
X6fHZhitX0N5TTEOY6iPtnPM9m5u872I054AhwO8HjpibQN1tHc0IPLaceYMXoHyzgYeJ+33HARp+P  
KTN+WtlriAR9NtR0QuwsTi1MSjRELd3I8uItbux4nERzmI4Eo1amXsBIpRryNnGn8MM5kgKTNTSRVQ  
W37XvzBi5S5voKkDMRTHN3y69SFuqk4dZpEsZ9YZTmRNTXcU1nFTyNdvPPMWoJtKZzFFmMXU9  
Q/4MRTeXVLq80prHmj0tPRdla4eSeDi9n//KPKeocIxkKMVbUiHfrFjb8pwdxVOR/CK1S05Hrber7rl27z  
P79+wsdxoooyGhvlbY+/jc9YjxDqSVF7Zw5LTSSTyMgI/PQr2Ip30XLuJNv6HVg2O/b1awlWN+OtKKK0  
OMFYNM5PjvVy18aKq1YS7RkoJRj2sHNzK05HkkvdVXT1VbBxbfdVL/gTZceCPrZvuoTHFaO1q4a+w  
VI2ruvCbru6ickYaO2qweeOEeq5+nhOR87yh6GaCI7UMdW6luOoCLu8YxkBkrIbQSB3rix7DWR5jpPY+x  
s7XkAy58K/pwR6L4AiGsY8GcA2N4OrrJ54ydK9bR8mOO0mWV+FvdGB3Zxn9FE9gnbnI2MFh+uLrW  
dfzTbzr3Pjq386x/lowaZLxNGUDJ7jvrWVUfvzXEEu77VR+iMgBY8yubMe0RqAub0ozcChB7b1OiESxzl  
7Cau/G6htERgMQiUEsjiQSEE+CMRi3AxxuTPUAUa8NeWgXFVWZtvgSABKAYe/5IRqrFOwK66pGE  
HE8PqxzViWweVM0Ly2C4cjNSXG2soRnPYu041kTaC3xOlae3UJAAgAutq+zjfUcdIbCvVpTaGujYT  
GqmjtO4MDmfkclvSS92Z5i2vv8HX+AC4Z51IIJ/XT/YbCSdfpJFfqrJgBgQIZHGT16noY9T+EaHsHU15  
Cqr4GSosx409EgVkcPVmcvpqaS0sZ6THmCFu9/waRsIXsbNlhp7LKRjJpeP3V2zj9/LM0nvo09Y98EbHr  
n6VaWvobpwgmglTH1pKOpyj68RPYT53H1FWRqqogXVsFG9dj3E5w2jF2O9jtmY7b889mZhKXrqNz4  
DA7i6bWJFr6gsSSadaVeaccE4GailEqSgOkUhZOR5KZphWUlwYoLQmQTltZE8BkNpthY0MPL3emKe9  
9kIQ7Qtmak9jsUxeRc3oClK89jtU7jN2bIF1TNn0cAqa8hOQt2/jhUJif31KJs38QGRyBrr5MGY+L9I3NJB  
+8G1yZ56QEg1R28GLby/zmrZ+4fDm7Xdi23c0x81bWdHyVni/8GXWf++MZvzelFpsmAkUoEeK2o+sp6  
TwOdU7iv/Tuyy9g00olYbQTGu8HwCk24umr38lHEikOXBrh1vWImXfK07Db0rO+sE+wBKwcy8YITLd  
vD1uKIjPeH8Cy0jjKUpSMvkA/72G2E2pK3PQEYhzpC3Hb+jWwbs2s8RS5/KQlxmhslBJXyeXHS8ss3G6  
L8Dt/g/T/+VPGnnmG4oceyuE7VGpxaIOkIrH7OWq71+PdXkH6tu2zJwGAsXZwl4A9U9YuNmLpyTNvD  
XvPD1Jf6qYoT8tKz6Y3MUSZrWjWJDAh4S7HYMMdbMup/Na6Ilr6gvSNRWYvDAhClaeKzmDHlGP1  
DTbOX7Io/9Vfpffzf0o6FMotaKUWgSaCVS52/jyuv3+MQNE6PA1z2KF8AL4ay5/6bDsxMyVZpfzfUFG  
I3GaqnjbhC4fehIDINiL53ROtLgR/8CRnMo6bRbb6kp46dwAseTUfo1sKjyVtI9N3f2sutqitzcF6zfjbG5m8F  
vfmlPcSi2EJoJVzBhD9x/+EV33PkjKPYbNnuMIsnQRtuhqPbyQw6xE0t1uwZiyTY3zrC9vqSJVLaDo9i  
cFMjWAO4p4qBIM7y7v2bCqlLnFT6XfysZRCTZR/ka1V5q2gLTk0EdrtQUWnR0ZGk5D3vYehb/0w6HJ5  
T7ErNlyaCVSz4/POkRobprW4GzxwWRBvtBHcR2K/UIOyWjWg6TjJteOFMPxur/XnbaSwXMRMjko5Sb  
Jt7jSRS1lh/4HDO5TfVFBGIJjjVNfu+BWXuUkKJEJHk1Bf5qkqLktJHLW1uDdvZuQHP5xD1ErNnyaCV  
coYQ//ff5nid72byEgx4gnkfvJgC/ivXrLBIXai6Th7zvbjc9tpKJ3fJjeLpSc+SJmteMZRSNOJe6sRk8Kd5Z17  
NpbAjvoSjnWMztpfIFhUeSrpCEyticVRW2ejuTpFOG/xvehMjjz029+CVmgdNBKtU5NBhUiPDuG66mVS  
gGIcttw7PzGihdi6OhE4LRtdYyFC8SRba3PvoM2X7sQgpXPsH5gsUtxEUX/ui8R5nDa21Zfw4tkBwrPsc  
1zpqaJtbGqHtMsleL1Cf18a19atpEZHi48OefYlZorTQSr1PCjj+K//35GRsHYozhdOb5yj7aDu/jyaKHL1wu  
mCMSj3NRQWtB+gQk9iQHK59g/MFnck9nNzDPWmvM51X4na0rdvHSmn3R6+iGu1d4qLgWyj0wqr7D  
R0ZIELAvvHXcy+qMfzylupeZDE8EqI5ECD73HL6776G/P03SNThlr5pDbZMqQ1cGgozOJbA4xEctsL/  
SkXSUaLpGEXz6B+46jolzfgHDiEm920qm6r8pDEcbBuZtky5u5zR2CjR5NQ1sysqhK7OzAgk7x13MPbk  
k7PugaDUQhX+r1YtueCLL+FisasJWWkpvb4qoYwCnLYe5A6k4jHaA/8poofahCBf7Q2woLyLFzE0iS6U7  
MUI5rWRe/QOTxd3lpG0uPCMtcrvxvoSLg6EaBvKPhfAEhtV3iras8xXKCuzGB1NE4sZHGVXAha7c2b  
uwSs1B5oIVqGxp3fjueUWAPr7UoQdPdhzqREMtYK34nKzUMdIhJa+IJtrivA4nMRn2PJxKXXH+ymzz7  
9ZaLJwSTP+waNIOvck57RZ7Kgv4dWWQYKx7M9JlaeK1tFLUx63LKgszKKnO4WI4LlpJ4Hnnp93/Erl

QhPBKmMSCUJ7XsZ7661Eo4Zo1BCSfpwWDjWCwXOX5w50j0Q41xNkc60fl8PCjo2EWQY1ApNJBO  
WOktnL5iDILCbpLsU3eGJO55V4Hayv9PLS2ez9BTXeatrGpiYCgPIKi86uzHPp3nkTwec1Eaj80kSwykS  
OHMFeXYWttJT+vhSlpRbx dBTHbE1D8RCE+qGolr5AjFM9QTbV+nE7bEBm+GjCJMlhTlVejaaDGMA  
nizd8NVLcjG/kNL YsY/9nsr4isx3nkY6RKcfK3OUEEkFCyanNRxUV1uV+AvfmzcRaWkgF5jC8V6k5yms  
iEJGHReSMiLSIyGeyHP8dETkpIkdkF5Kcisj6f8SgI7tmDe9uNAPT1pSguERKpBE7bLOsPDraAv4bBSIrjn  
WNsqvbjGU8CkFIJ1CYWiQL3E3TH+6lwICzq8NWU3U3MX09R/8E5n3vjmhLO9k5dj8gSixpvDZeyNA/  
5/UIiYRgbSyNOJ65Nmwi/9tq841dqNnlLBCJiA74CvB3YBnxIRLZdU+wQsMsYsxP4HvBX+YpHZQT3v  
Iz7xkwi6O1N4StOYbPsyGy/CgNnCXlqOdI+SIOVD6/LNqWIHXvB+wk6En1U2BenWWiySFEjrlA3zsjA  
nM5z2jPrEe05N0A8dfV6RDXealpHL045R0SorLTR2THePHTDDYReeXX+wSs1i3zWCO4AWowxF4wx  
ceBR4N2TCxhjnjfGTNS39wINeYxn1UsFAsQvXMC1cSOplGFwMI3bH599xFCon3Qiyr4eWFfupcidvfbg  
EAdxU7hEkDJp+hNDlNkWPxEYy0a4ZCPFva/P+dzKlidlPievXxi66vFaXx0Xp5mnUFll0d6eSRyuG24g9  
Prc76tUrvKZCOqByXP0O8Yfm85/Bp7KdkBEPiEi+0Vkf39//yKGuLqEDxzA1dyMOJ0MDqbx+YSURGft  
KE71naY9XU51iYcy3/Sji5yWjZiJT3s83/qTQ/htXpx52hw+5lsDJolndG7DSQE21xTRMxbloqQhpX6nH0s  
s+rPUMioqLPr6UiQSBuf69SS6ukgODy8ofqWmsyw6i0XkI8Au4IvZjhtjvmaM2WWM2VVVbW0wa0g4  
ddex7V5EwC9PUnKyiyiqdiMNYJ0KkG6v4WQq46a4pmXqS5001BHRJfyPNQGJguVbaW4/yC21NwSns0  
Stq8pYe/5QLjs1AIQp2vlosjF6aUdzgyw0g7O1KI3Y5rUzORQ4cX41tQaop8JoJOYO2krxvGH7uKiLwF+  
EPgXcaYWb7jWfXC+/fj2rQZgO7uNGVIFpFkBNCMieD8meOELC81IWWzXt9p2YkWsEbQkeij0l6a13u  
knEXEPDX4+/fP+dwSr4P6Ug+vnL+yZHWNr5bzo+ezlq+qtrhwMZNYXU1NhA8emH/gSs0gn4lgH7BJR  
BpFxAl8EHhicgERuQX4Kpkk0JfHWFa9dDRK7Nw5nBs3kk4b+vpSIJVnEsF0NYIzPWOUBVugNLfBX  
HaxEytQjSCYihBNxylZ4LISuYiUNOMKduEM98753MYqP2ORBOf7ggDUemrpCfUQS019D1RTY607  
K0U8bnBt2kxk39yTj1K5yFsiMMYkgU8Cu4FTwHeMMSdE5PMi8q7xY18E/MB3ReSwiDwxzeXUAkWP  
H8fR0IDlctHfn8brE5xOIZKM4MjSR9A3FuF8WwelVpiUtzqneziwX7Nd5dLpTPRSZS9dkIVPjWUjXLaf  
kp5XkHRuO5NNsAS215dwoHWYYCyJ3WanyltF62jrlLJOp1BeYXHxYhJnUxPRM2cw8cLVuNTKldc+  
AmPMk8aYzcaYjcaYPxt/7LPGmCfGP3+LMabGGHPz+Me7Zr6imq/IkSO4mpoA6OxIUIGRGf4ZSUZwX  
bOSaDie5IWzA9zh6yHuX0Oui/Y4xF6wzuL2WA+VjtmbrxZL3FNFyllE0cDc5xb43XbWIXv5WcsAxhjqf  
HWcGzmXtWxDvY1TpxKI2429pobombMLDV2pKZZFZ7HKv/CBgzjHE0F7e4rKisyLeyQRuWrUUDqd  
5oUzfawrcVAeaSPuy31Eb2bf4qVPBEmTpD85TEWeO4qvFS7dgnusFWe4Z87nbqj0EUukONMzRoO/ngu  
jF0hnWeW0otIiETf09qZxNTUROZLbfspKzYUmgIXAGJOpETQ3EwymCYXS1JZlfvSRZBiX7cpooInlk2  
9w9JBwlZOyu3O+T6Gahrri/ZTYirBbUye55VPachAu20ZJ98+w5jiKCMmsUnq4fZRE0oHf7qMjyz7JIsL6  
DTaOHonjbGzUDmOVF5oIVoFkby8mmcRWWUlW5KqahuWNV4jSEZwjieCjuEQFwZC3Fhfgn/kNDH  
/2pkuO4XDchArwISytnh3pn+gAOKeChLuSop7XpnzuV6njaYqHy+3DLDGX8+ZoezLTdfX2xgeTjNWuZ  
nosWMLDVmpKTQRrAKRo0czE8IEuHA+SU31IR97JBXFaXMRiiX5WcsgO+qLKyp0A5Bwl87pPg6xE  
1/iGkHapOmI91PlKF/S+04WKdmElz6Kd3ju+wasLfeCgUi4lLNDZ8m2ap9lCY1Ndo62lZAcGCQ1MrLwo  
JWaRBPBKhA5chTn+vUEAmnGAmkqKq/82KPJTb/BnnP9rC3zUup14hs6SdS/bs73cYqd+BKvQNqbHM  
Jrc+POZRntPDGWRbB8J0UDh3FEB+d8/ra6Ylp6UtgsR9ZN7QHq6y3GxgzBzfcSOT63JbGVmo0mglUgc  
uQwzsZGzp5NUFd3pVkJIJaKc7Y7TDJtaKz04YgNY4+PEPfWznDF7EQEm9iIs3TNQ5di3VQXqFlosptD  
Q6hsK6WdL8551rHbaWNTtZ9ouJQTg9k3q7csowmjnXMV9xM+enQxQlbqMk0EK5xJp4mdOo19QxPnz  
iZpaLjSoRpNRrGwcaYnxPY1maWbfYMniPnXYua5z6NLHETSU/fizQtjaIt1U13AZqHJ4t5qEp5qSrpehD  
nuM7ymzIObGk4MnM46egigrs4iavnpOjplgr5SC6KJYIWLX7yIVVRE+4ALr0/w+6/8yEeiAVIpO1vrinE5  
LGzxIK5QJ9E5dhJP5rIcRNJLs1JIT3IQI+XAay3eJjQLFS5pRtIJiuexd8H2uhqSSRcHurL3NViWsG6tcDow  
/5+PUTloIljhIseO4Wxs5MSJBOvWXT288tnT7TgsN1Xji8n5h04Q89VjFjAM0ylOokvUYdwa66JmmdQGL  
hMIVezAFWjFk2WvgZk47RY1nnr2XDpAMsv2lgANG/2MeuoZPNWe9bhS86GJYIWLHD1KoG47kbCh

etJooWOdowyERyl1ewGwJSO4A61Ei+beSTyZU+xL0jSUNmkuxbqpcVTm/V5zlbYcBCtvorjv9TlvZLOpt  
JG0vY+nT2TvNLbZhep0F0ef0hnGavFoIljhokeOcp4trFtvR8bb/YfDcXYf76Gp2oHTytQGfEPHiXlrSc+2S  
c0sHLI0TUPdiX68Njcea+alsQsl5fATKt9GaecL2OJT9yWejt3mpMJdw8mhU1wcyH5eTUmE8+eTpFPZa  
w1KzZUmghXMxOOMtA/RG/BQX5/5UafShscPdLK9vgSxRXHYXNhSETyjF4gWbVjwPV1LIahaou3U  
2pdfbWCyuKeKaNE6yjt/ipXKfSTVGu96PEVt/PuhTiKJqYvaldaX40wEaT+tG9WoxaGJYAWLnj1HV9N  
D1NfbNsztYHnTvdhtwk31BYRSYVxWi58g+O1AfVc3107rfyPGkqYBF2JA WodFXm9z2KIFq0j4SymtO  
sFZJrRQNcqc1WRljhVZRH+40jXIOP2ulpKBk5xbt/cl8FWKhtNBCtY6MhROktvpmFtpvO3pS/I8Y5R7m6  
sQASiyRBuI5naQHHjotzTKQ4ied5f6EKskwp7KY48bUm52MKIN4BJUZLrMhQi1HnWY/e2MhCIs6/16r  
2OLZ+P8mgHFw/1kUpq85BaOE0EK1jL/l78zgQ+n8VYJMG/H+7i3uZK3M7Mjz2ailASuETMt2bBfQMT  
3JaLSJZNVhbT2Wgb9cuwk3haAqHyHdhjIxT35rYJfZ1nA13hi9y50c+LZ/vpGI5cddxbW4rXmaTr7EgeAla  
rjSaCFezCcBn1a4RU2vC9Ax1srVTU3KI+SeaDFIS7CZavGHR7ukSB3GTJJVjM8hcDSVHiKZilC+D2cR  
zYSwbgcqbcYV7KO7bN2t5h81JpaeOvvgZ7mq54Lv72hkJXxmW66ipoTTVR8sh3dhPLZwmghVquLWfg  
L2CuuZSdp/owWYJW9dcWa8/bdLEUxHwryNtORbvxgIey0k4HZm97DycilyiwVWd6145y4qx7IxV3YIr1  
E1Jz6uz9hk0eJs4N3aE+jIX2+qL+fbeS4xFMp3O9jVrKOo+TuvRzOY2Si2EJoIV6uSTJ6hIdXO0e4yWviD  
3NFcwaYkhUmPn8BiI+3Pbj3guPJaLUB4SQczEaYt3U+/IbevM5chYDsaqbsMeH6Ws/afYUtN3rPscJfjsxb  
QGznBDbRHNVUV842cX6RgO46itxd55FhEY7AwRS6Z47nQvf/nUaT73xHF+crybVFoThMqNJoIVyBh  
Dy+kIPleQZ0/1cf+Wapy2ST9qA7auF/HYPBhr8X8FnJaTYB4SwdlIG9WOcypLWYMPAGPZCFTcTNrho  
aL1R7izbEgzYa2vmdMjB0gbw7Y1RexaX86j+9r5zqEeov5SklaYr/zLUW7/wrN86emz9AdjxJKGLz19ll/8+  
muMhp+fw1/bk+hl2oORnsDBKPJjkUinLPLRWUeq7+MfsGjtApKZz2/Gzt6BYnodTiJoKUSXMqcoFbf  
Dcs6nULRjLrEiVc5RT1vo535AyBiptIeK7uBC91VmFZdjpC51nnb2ZtuYfaknpaB0L0+8pJDLVRkt7Cn39  
oBxX+K/0/79xRxzdfbeXX/+UA3/7Pd+Cw6Xs+NT397ViBDu/poHjgNJWbNIBfdvWCbJKKUdb5Uwb89b  
jytIa/W5wEU+FFvebFeCdFNh9FNu+iXrfQE u5yxmr JuksoazzBSouPYlntOXK1pcCG3ybOT78GunxvgC  
HTdhU46dm0wa2Ri7iCqUosV29PpRICR+9ewOReJKvvnh+qb8tdZ3RRLDCjITjHN3TjjfWxca1U4dYlrc/  
S9zXQMACt+RneQav5SaQyn1ZhdmkTZojobNscNYt2jWXEyNC1N/ASN0biPnX4hm9QNWfxylviewbf8  
CkqxY1NbLSFr15fKF27BnvbeTzVbsZaA1Oua1nCx+5t5J/2XKR9aHETs1pZNBGsIKORBL/55b240imKq  
qe+c3YF2/GMnCFQeQuBVCRvu3p5bR7GU sFFu96FeCcey0mZo3jRrrksSWZZimDlTk bq3kjcV4sz1E1F+  
zPcPNbPyb7ncQ2dRMbnaRifD5xOfN4Eoy1jWS9ZXezmLVur+Ztn dJE6NT3tI1ghBoIxPvL117g7ZlGV6I  
C6+qsLpFNUXPwRgarbSNucjKWCbHQ35CUWtzhJkiJuEjh lYR27KZPmcPAMO7zNixTd9cFYNuKeKu  
KeKgA8iSC+cAsX+l/m7os/Iu6tJlyymVhVBcXBS1wcXotJG8SaOq727dvr+PR3j3C+P8jGKv9SfyvqOqA1  
ghWgpS/Ie77yM7avKaZuLE1Z+0HS1ySCso7nSDmLiPo3kDJpIuk4HnHnJyABv+VldBFqBScjFyiyeSmxr  
+4XsJTDz8aiLbzuhEtN7yRcu hI XuIvi9Dkq9z+GR0bg+GuQTk451+ey89ZtNfzjC9pXoLLTRHCd+8nxbt7/  
v17hnTvreHhNBZJM4nbEMd4rTUOusUv4ho4xVn07CARSQXyWO6+Tsrw2z4ITQSQd5UTkPJs8C9sjYa  
XwWR7qnVXsD58h5qtnrPp2xrY/TGzUQZmvn8Tre9n13dtpeuX3KOI+GdJXVi59aFsNu0/00Du2RNuIquu  
KJoLrVDCW5PceP8rnnjjJpx/awv2bqyk6NUyJ6Sddf2UrQysRoerCDxmruZu0LVMDGE2F8Fp5qg2M81se  
hhPZ261z9XrwOGudNXmP9XrS5KynOz5Id2J8wxuPG+P34/K7OZ+8n4t3fB5jc7Jh3+e57fF7WHfgf+Aeu  
0CR28G9zZV885XWgsavlidNB NcZYww/Od7Dg196gf5AID9773aaq/2YtGH41AhI XYdIN6yfKEzlhceJFK  
0j5ltz+RjqQBeW35fXIstH/3J+a+X3xbrZiA5wgbXmtkLryI2y8ZWTyOvBA6TMOPNQLVV+LrOkUwII/  
FqBte/g4t3fJ5Lt/wuznAP2596P9t2f4iPlh3nsb0XCcenNh+p1U0TwXXkeOcoH/zaXv78yVN8/L4mPn7fRrz  
OTH9/oC2I3QHe3gukazPDLEs7nseWihGsvPmq6wwmRymy+flaa4nNz0hjqPQ8Fp8Lp6PsDR5lu6cZm+iv  
6LUqHaWU24t5PXgMAFN Xg7RcpKQ6Rf+IK/MJ4v56+jZ9iHP3/R2B6l3sOPdlnrJ9iqPffwQS+V kLSI2f9  
K/sOtA+FOa3/u0gv/yN17mxvpg/f+8Oblxz9azgwWODlFoDpNZtAJsNX/8R/EPHGKI7A1zzYjqYGKHYlt/  
OV5tlwyMeRIJT x7fPJG3SvDC2n3Wu2lXfQTyTza719CQGuRDrhIoyiEQp9YzRe9HGtWvQGcvOaN29tO

76Yy42/wqesz/E/M12eOkRiIwUJH61vGgiWMZGwwk+/x8neMff7cFls3jk/Tfx0NZabNcMEUxGkoxeCFB  
+8VVS6zfiGW2hrPNZhurfdL1fYEIkHSVJCm+eJpNNVmr30x2fw+btBl4NHsWGxQZn/ezIVzGbZeMm72  
ZeDx5nKDUGDXV420+SSgjB4WlGA YhQtO4m/tXzSxy94VNw6RX4nzvhdy+FgZYljV8tL5oIlqFU2vAv  
e1t54JHnaR8K85fv28n7bluLx2nLWn7g2BD+ShuO3nYcZULFhX9ntO4BUS6pawkNJEYosflhCZZxrrKXc  
SnenVthAwfDpxhMjrDds3FJ4rve+W1etnoaeW70dcINVVgnzlJWI6L73MzTg+7eWMG3zwjpez8F7/xbiAz  
B/34Ivv4WeO2rMHSBKdUKtaLphLJ15njnKL/3+FGMgd97+AbWV8zclm9Shv6DAzSYM8jaGiovPs5o3X  
3EPdl38OpM9FNmX5oZuuX2Eo5FWoiko3hmGvljDPtCJ+hM9HOrdys2K3vCU1PVOMqJpeP8JHaC9yK  
USzfnWhtoujWBfZq5fJuq/ew518/+1iHuaKyCW38FbvowdB2Elmgxb8CywZ1N0H1NqjcDBUbobwJfFVc  
15tBqBlpIlgmIvEUX3r6DI8f7OADt6/ljZuqkBz+4IZOD2N3W5Tu2U1pc5DRujcS99ZkL2ygM97LTd4tixx  
9dpYIdY4KTkUucqtv9YYCZPk5cAhgukIt3m34dAkMGfrXLWkSXO4rosbD76Kf/P76G6xsXZrKmt5EeH  
+zVX862tt3Lq+HLsIYHPA2jszH8ZAsBcGW2CsA44fh0AXjHVmjIVuhrqdmURRuwoqtoJzZS0GuNpoIlg  
G9pzs5/e/f4zGSh9/8Qs7KfbktixDOpWm6+UemiIv4fRECDS/iYSrfNryw6kxjDH4Lc+0ZRbbetcaXgseZ4t7  
Az7b1fftTQzys8BhyuzF3ObdiqXvNOdtg2sNPRssks9fJLHzNO0ntrFmUwrbNH/hG6v87Gsd5qnj3fz8zmu  
G6IpAUW3m41rRURi+BMMX4fSP4JUvw0gb+KsyNYaSdVBcB/4a8JaDtztIyiqBU+Z1iaWKU0EBdQ7  
FuVPf3SSfa1DfPSeDdy8tmxO5/e90k5J8hK+I68Rf+PdpGdIAGBnopeoc1Ytafu7x3KxwbWGnwZe5w3+W  
yix+elPDnEqcpH+xDBbPRuocswct8pNrbeWwOYInlefo/fGMs4eK2brLdlrWCLC226s5ZvtHLr+jLqS3J8c  
+AuGa8N7LzyWDofgZ5MrSE0AKOd0HcKYmMQC0BkOPO4SUPpeqjclGlyqrkx81HWCHnYIEnlThNB  
AYxFE3z9pQt885VWHTxaw1/8wk7cjjk2iRx9hcHXDVv6nyfd2ISpmKY5aFw0HaM11sld/p0zlsuHDc46bA  
jPjb5G1CQosnmpc1Sy2b8em74ALKr0xg1Ud4wxEH6dttNv5XDRHh5cfyPVWZJtuc/JA5ureGT3Gb7wnu3  
4nPn8ObBsUFKf+ZhJPJhJGKMdmQ7piy/C0MVMwqjaAjXbM4mh6obM10V1WoNYIpLPja9F5GHgfwI  
24OvGmL+45rgL+GfgNmAQ+IAxpnWma+7atcv378/PwHnWUtfkEf3tfHd/R3cvLaU991aT1XR3Gb4ekZ  
bKPnZ13n+1DtYkzpBUcdZzJvuhVl2oHo5cAiDYZNb1+1Z6WzBCJUvHef8PQ8wHNrAj278MpWlbu713  
8xtvq2U2ouuKr/7RA+DwRifeftWitwFeG8YD8JQK4y0ZpLEaAeMtmcmvZU3ZTqqKzZBRfN4p/XGTLO  
TJok5EZEDxphdWY/lKxGIa04CzwEdAD7gA8ZY05OKvObwE5jzG+IyAeB9xpjPjDTdZd7IjDGEE2kG  
YnE6RmNcqE/xJGOEV4+N8BIJMG9Gyt4y9YaqtzTwBWIkh10tUnnuc9tYSXh/7AGsCRykbPIW5705  
wzb6vwPeHfspNvs34lrB/QBWOCyhI2Wun6dz6BobsWxnddoSjZa9wPtZBhaOU7Z6NbHavp81VT5W9jJf  
ODnC8a4xfvnsDdzaVY1sOL7LxYKaZaawLAp0Q7Mt8PtqRSQKl6zMfZRugdG2mBuGvyfRJeMvBXapN  
TpMUKhHcDXzOGPO28a9/H8AY8z8mldk9XuZVEbEDPUCVmSGohSSCrpEIPz3dB8ZgyIzXT6UNiZQ  
hnkwTTaaIxFMEokkC0QSjkQQjkcz/gUiCUDz7KIyZ1JW4WV/hZWtdMRur/FOa5x2JABu6n8SWjmNLx  
bCnIjiTY7hjA/jDHZQFznG04wEOm/cRdKzBloxS3/USTneMZH0VZFI/PpvDqTaqxI9DWwNXDXsiSXlb  
gKRVRUf9/SAWrlQr3Rt3c3jNID1mlDGurEbqN25sMSfOlIMiy0Wxw8X6Ej8lbid2sbAhgGAJCIKFZP6Xi  
c/BJtakx8iUR5Dxc678tmY+a3SVcbtvLXNiTKbvIdADwR4I9UN4AMKDEB6GdGKaJ8QFDl9mhJPDBw  
432N1gc2WO2V2Z0VO2if+dmf8te+Z/sWWawcQan60vk/rbBBh/2TLj/5j0pI9rXtImrmFNXHPS/xOPXY4r  
4/cS2PIOkj5GXg6MyWCfL4q1APtk77uAO6crowxJikio0AFcNV0VBH5BPCJ8S+DIjJ4bZlc2Iloq62y+0o  
WtYmZM2pjxn2wmYZnJBzGYyQ91AQdmuFyJFbGttw3O+JZ+JPIacfJhDTWNevNB2IBilxF05w5Kbj18  
AYv3wLRAEXu2Z+L1eCq58KA8C+kLQcGQdLdjPqvFimBZJy1S/ykrEMNIbjeVn4SAQZjhhtU+wBB  
sEybRAX7+6g6ajK2B653n6+ukOXBdvD40xXwO+NvG1iOyfLrOtNiKyfyA4oM8FmediUJ8LQJ+LCSky  
v2Msveqfh9nkswGtE5hc52sYfyxrmfGmoRKu95StlFLXmXwmgn3AJhFpFBEn8EHgiWvKPAH8yvjn7we  
em6l/QCml1OLLW9PQeJv/J4HdZIaPfsMYc0JEPg/sN8Y8Afxv4Nsi0gIMkUkWufja7EVwdx0urtDn4gp9  
LjL0echBXucRKKWUWv50kK1SSq1ymgiUUmqVu+4TgYh8WkSMiGRfgH8VEJEvishpETkqIj8QkdJCx  
7SURORhETkjIi0i8plCx1MoIrJWRJ4XkZMickJEfrvQMRWaiNhE5JCI/KjQsSxn13UiEJG1wFuBtkLHU  
mDPANuNMTvJLOvx+wWOZ8mML2XyFeDtwDbgQyKyrbBRFUwS+LQxZhtwF/BfV/FzMeG3gVOFD

mK5u64TAfA3wO9SmEmRy4Yx5mljzMSU471k5mysFncALcaYC8aYOPAo8O4Cx1QQxphuY8zB8c8D  
ZF4AV+3mzyLSALwD+HqhY1nurttEICLvBjqNMUcKHcsy8zHgqUIHsYSyLWWyal/8JojIBuAW4LUCh  
1JIf0vmjWK6wHEse8t6iQkReRbIsk0Sfwj8AZlmoVVhpufCGPPv42X+kEzzwL8uZWxqeRERP/A48CljzF  
ih4ykEEExkn0GeMOSAiDxQ4nGVvWScCY8xbsj0uIjuARuDI+L6+DcBBEbnDGNOzhCEumemeiwi8lH  
gncCDq2x2di5LmawaIuIgkwT+1Rjz/ULHU0D3Au8SkZ8D3ECxiPyLMcYjBY5rWVoRE8pEpBXZYyZ  
84qkK8H4BkB/DdxvjOkvdDxLaXyNqrPAg2QSwD7gw8aYEwUNrAAk867oW8CQMeZTBQ5n2RivEf  
3Y8w7CxzKsnXd9hGoq3wZKAKeEZHDIVK/Ch3QUhnvJJ9YyuQU8J3VmATG3Qv8EvDm8d+Dw+PviJ  
Wa0YqoESillJo/rREopdQqp4lAKaVWOU0ESim1ymkiUEqpVU4TgVJKrXKaCNSKJiK1IvKoiJwXkQM  
i8qSIbJ7ntb4pIu8f//zrEwu6icgfXFpuD8dX/zw6PoTzzoV/J0rlz7KeWazUQoxPsPoB8C1jzAfHH7sJqCEzCQ  
0RsU9asC9nxphfm/TlHwB/Pn69u8nM8L7VGBMbXx7ducDvY14xKpUrrRGolexNQMIYc3mC3fgihTYR  
2SMiTWApx9es/6KI7Bt/F//rkEkkIvLl8b0OngWqJ64jIi+IyC4R+QvAM/7O/1+BOmDAGBMbv9+AMaZr/J  
zbReQVETkiIq+LSJGIuEXk/4jIsfF18980XvajIvKEiDwH/FREfCLyjfHzDo0vuqjUotAagVrJtgMHpj12K5k  
9HC6KyCeAUWPM7SLiAn4mIk+TWb1zC5l9DmqAk8A3Jl/EGPMZEfmkMeZmuLzg22dF5CzwLPCYM  
eZFEXECjwEfMMbsE5FiIEJmvXxjjNkhIjcAT09quroV2GmMGRKRPweeM8Z8bHzjoddF5FljTGgxniilu  
mkiUKvV68aYi+OfvxXYOdH+D5QAm4A3Av9mjEkBXePvzmdkjAmKyG3AfWRqJI+N75p2AOg2xuwb  
LzcGICJvAP5+/LHTInIJmEgEzxhjhif+C4R+W/jX7uBdeimK2oRaCJQK9kJ4P3THJv8TlqA3zLG7J5cY  
L7r9IwnjheAF0TkGParTF8zmcm1Mb7PGHNmPjEpNRPtI1Ar2XOAa7zpBwAR2Unm3fpku4H/Mr6EMy  
KyWUR8wEvAB8b7EOriVMPPIjHp3C0ismnSsZuBS8AZoE5Eb8vVzS+cuoe4Bcn7kvmXX62F/vdwG+  
Nd4AjIrfk+BwoNSutEagVyxhjROS9wN+KyO8BUaAV+OE1Rb8ObCCzp4UA/cB7yIw4ejOZvoE24NVpb  
vU14KiIHCSzHPjfj7fj4EW4BPGmLiIfGD8mIdM/8BbgH8A/nG85pAEPjo+2ujae/wpmR23joqIBVwkMz  
pJqQXT1UeVUmqV06YhpZRa5TQRKKXUKqeJQCmlVjlNBEoptcppIIBKqVVOE4FSSq1ymgiUUmqV+  
/8BmgJ4JfoAnXAAAAAAASUVORK5CYII=\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": {
```

" i m a g e / p n g " :

"iVBORw0KGgoAAAANSUhEUgAAAYsAAAEGCAYAAACUzrmNAAAAOXRFWHRTb2Z0d2FyZQBNYXRwbG90bGliIHZlcNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXBIWXMAAAsTAAALEwEAmpwYAAAUX0lEQVR4nO3df5Bd5X3f8fcHMP4dJGCjEkmumFglwWmNyQ6QuP NJrUb8aGMxGUNx66IStcpMSBonrRvcdqoEwtSeuKXg1nTUICw8KT9MQIFTaqrKdj1pzY/FEMyPUDbYGGkAbZDA2NSkYr794z5rX8SuzrW8Z3fFvl8zd+453/Occ56dkf3hPOe556SqkCTpUI5a6A5IkY/w0KS1MmwkCR1MiwkSZ0MC0lSp2MWugN9OPHEE2vNmjUL3Q1JOqLcd999f1ZVYzNte12GxZo1a5iYmFjobkjSESXJk7NtcxhKktTJsJAkdTIsJEmdDAtJUifDQpLUqdewSPJrSR5O8ICSG5O8KcnJSe5OMpnk5iT HtrZvbOuTbfuaoeN8tNUfS3J2n32WJL1Wb2GRZCXwj4DxqvoJ4GjgIuDjwFVV9U5gP7Cp7bIJ2N/qV7V2JDm17fcu4BzgU0mO7qvfqkTX6nsY6hjgzUmOAd4CPA28D7i1bd8OnN+WN7R12vZ1SdLqN1XVy1X1NWASOKPnfkuShvQWFIW1B/gE8A0GIfECcB/wfFudaM12Ayvb8krgqbbvgdb+hOH6DPt8V5LNSSaSTE xNTc39HyRJS1hvv+BOspzBVcHJwPPAZxkMI/WiqrYCw wHGx8d/4Dc6/eRHbviB+6TXn/t+5+KF7oK0IPochvobwNeqaqqq/h/wB8B7gWVtWApgFbCnLe8BVgO07ccBzw3XZ9hHkjQP+gyLbwBnJXILu/ewDngE+ALwgdZmI3B7W97R1mn bP1+Dd77uAC5qs6VOBtYC9/TYb0nSQXobhqqqu5PcCnwFOADcz2CY6L8CNyX57Va7ru1yHfCZJJPA PgYzoKi qh5PcwiBoDgCXVtUrffVbkvRavT51tqq2AFsOKj/BDLOZquo7wAWzHOdK4Mo576AkaST+gluS1MmwkCR1MiwkSZ0MC0lSJ8NCktTJsJAkdTIsJEmdDAtJUifDQpLUybCQJHUYLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0lSJ8NCktSpt7BIckqSB4Y+30zy4STHJ9mZ5PH2vby1T5JrkkwmeTDJ6UPH2tjaP55k4+xnlST1obewqKrHquq0qjoN+EngJeA24DJgV1WtBXa1dYBzgbXtsxm4FiDJ8QxezXomg9exbpkOGEnS/JivYah1wJ9W1ZPABmB7q28Hzm/LG4AbauAuYFmSk4CzgZ1Vta+q9gM7gXPmqd+SJOYvLC4CbmzLK6rq6bb8DLCiLa8EnhraZ3erzVZ/ISSbk0wkmZiamprLvkvSktd7WCQ5Fng/8NmDt1VVATUX56mqrVU1XIXjY2Njc3FISVIzH1cW5wJfqapn2/qzbXiJ9r231fcAq4f2W9Vqs9UISfNkPsLig3xvCApgBzA9o2kjcPtQ/eI2K+os4IU2XHUnsD7J8nZje32rSZLmyTF9HjzJW4GfBX5xqPwx4JYkm4AngQtb/Q7gPGCSwcypSwCqal+SK4B7W7vLq2pfn/2WJL1ar2FRVd8GTjio9hyD2VEHty3g01mOsw3Y1kcfJUnd/AW3JKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhIU nqZFhIkjoZFpKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhIU t7aJsk1SSaTPJjk9KHjbGztH0+ycfYzSpL60PeVxdXA56rqx4B3A48ClwG7qmotsKutA5wLrG2fzcC1AE mOB7YAZwJnAFumA0aSND96C4skxwE/DVwHUFV/XIXPAxuA7a3ZduD8trwBuKEG7gKWJTkJOBvYWVX7qmo/sBM4p69+S5Je q88ri5OBKeD6JPcn+d0kbwVWVNXTrc0zwIq2vBJ4amj/3a02W/1VkmxOMP FkYmpqao7/FEla2voMi2OA04Frq+o9wLf53pATAFVVQM3Fyapqa1WNV9X42NjYXBxSktT0GRa7gd1VdXdbv5VBeDzbhpdo33vb9j3A6qH9V7XabHVJ0jzpLSyq6hngqSSntNI64BFgBzA9o2kjcHtb3gFc3GZFn QW80Iar7gTWJ1nebmyvbzVJ0jw5pu fj/wrwe0mOBZ4ALmEQU Lck2QQ8CVzY2t4BnAdMAi+1tlTVviRXAPe2dpdX1b6e+y1JGtJrWFTVA8D4DJvWzdC2gEtnOc42YNucdk6SNDJ/wS1J6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhIU nqZFhIkjoZFpKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhSerUa1gk+XqSryZ5IMIEqx2fZGeSx9v38lZPkmuSTCZ5MMnpQ8fZ2No/nmTjbOeTJPVjPq4s/npVnVZV069XvQzYVVVrgV1tHeBcYG37bAauhUG4AFuAM4EzgC3TASNJmh8LMQy1AdjelrcD5w/Vb6iBu4B1SU4CzgZ2VtW+qtoP7ATOme c+S9KS1ndYFPDFk9yXZHOrraiqp9vyM8CKtrwSeGpo392tNl v9VZJsTjKRZGJqamou/wZJWvKO6fn4f62q9iT5YW Bnkj8Z3lhVlaTm4kRVtRXYCjA+Pj4nx5QkDfR6ZVFV e9r3XuA2Bvccnm3DS7Tvva35HmD10O6rWm22uiRp nvQWFknemuTt08vAeuAhYAcwPaNpI3B7W94BXNxmRZ0FvNCGq+4E1idZ3m5sr281SdI86XMYagVwW5Lp8/ynqvpcknuBW5JsAp4ELmzt7wDOAyaB14BLAKpqX5IrgHt bu8ural+P/ZYkHaS3sKi qJ4B3z1B/Dlg3Q72AS2c51jZg21z3UZI0Gn/BLUnqZFhIkjoZFpKkToaFJKnTSGGRZNcoNU nS69MhZ0MleRPwFuDE9huHtE0/xAyP3JAkvT51TZ39ReDDwI8A9/G9sPgm8O/665YkaTE5ZFhU1dXA1Ul+pao+OU99kiQtMiP9KK+qPpnkrwJrhvep qht66pckaREZKSySfAb4UeAB4J

VWLsCwkKQIYNTHfYwDp7ZHckiSlphRf2fxEPAX+uyIJGnxGvXK4kTgkST3AC9PF6vq/b30SpK0qIwaFr/ZZyckSYvbqLOh/mffHZEkLV6jzoZ6kcHsJ4BjgTcA366qH+qrY5KkxWPUK4u3Ty9n8Oq7DcBZfXVKkrS4fN9Pna2B/wycPUr7JEcnuT/JH7b1k5PcnWQyyt1Jjm31N7b1ybZ9zdAxPtrqjyUZ6bySpLkz6jDUzw+tHsXgdxffGfEcvwo8yuDhgwAfB66qqpuS/AdgE3Bt+95fVe9MclFr97eTnApcBLyLwTOq/keSv1RVrxx8IkISP0a9svi5oc/ZwIsMhqIOKckq4G8Cv9vWA7wPuLU12Q6c35Y3tHXa9nVDQ143VdXLVfU1YBI4Y8R+S5LmwKj3LC45zOP/W+CfAtP3PE4Anq+qA219N9971PlK4Kl2vgNJXmjtVwJ3DR1zeJ/vSrIZ2Azwjne84zC7K0mayagvP1qV5LYke9vn99tVw6H2+VvA3qq6b0562qGqlbVeFWNj42NzccpJWnJGHUY6npB4N7Bj8C/JdWO5T3Au9P8nXgJgbDT1cDy5JMX9GsAva05T3AaoC2/TjgueH6DPtIkubBqGEExVIXXV9WB9vk0cMj/fK+qj1bVqqpaw+AG9eer6u8CXwA+0JptBG5vyzvaOm3759uDC3cAF7XZUicDa4F7Ruy3JGkOjBoWzyX5UJsGe3SSDzH4r/7D8RvAryeZZHBP4rpWvw44odV/HbgMoKoeBm4BHgE+B1zqTChJml+jPhvqF4BPAlcx+CX3/wb+/qgnqaovAl9sy08ww2ymqvoOcMEs+18JXDnq+SRJc2vUsLgc2FhV+wGSHA98gkGISJJe50Ydhvor00EBUFX7gPf00yVJ0mIzalgclWT59Eq7shj1qkSSdIQb9f/w/zXw5SSfbesX4D0ESVoyRv0F9w1JJhj8VgLg56vqkf66JUlaTEYeSmrhYEBl0hL0fT+iXJK09BgWkqROhoUkqZNhIUnqZFhIkjoZFPKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpU29hkeRNSe5J8sdJHk7yW61+cpK7k0wmuTnJsa3+xrY+2bavGTrWR1v9sSRn99VnSdLM+rYYeBl4X1W9GzgNO Cf JWc DHgauq6p3AfmBTa78J2N/qV7V2JDkVuAh4F3AO8KkkR/fYb0nSQXoLixr4Vlt9Q/sUg8ec39rq24Hz2/KGtk7bvi5JWv2mqnq5qr4GTDLDO7wlSf3p9Z5FkqOTPADsBXYCfw08X1UHWpPdwMq2vBJ4CqBtfwE4Ybg+wz7D59qcZCLJxNTUVA9/jSQtXb2GRVW9UIWnAasYXA38WI/n2lpV41U1PjY21tdpJGlJmpfZUFX1PPAF4KeAZUmmX7q0CtjTlvcAqwHa9uOA54brM+wjSZoHfc6GGkuyrC2/GfhZ4FEGofGB1mwjcHtb3tHWads/X1XV6he12VInA2uBe/rqtyTptUZ+rephOAnY3mYuHQXcUIV/mOQR4KYkvw3cD1zX2l8HfCbJJLCPwQwoqurhJLcweKXrAeDSqnqlx35Lkg7SW1hU1YPAe2aoP8EMs5mq6jvABbMc60rgyrnuoyRpNP6CW5LUybCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0ISJ8NCktTJsJAkdTIsJEmdDAtJUifDQpLUybCQJHUyLCRJnQwLSVInw0KS1KnPd3CvTvKFJI8keTjJr7b68Ul2Jnm8fS9v9SS5JslkkgeTnD50rI2t/eNJs52TkISP/q8sjgA/OOqOhU4C7g0yanAZcCuqloL7GrrAOcCa9tnM3AtDMIF2AKcyeB1rFumA0aSND/6fAf308DTbfnFJI8CK4ENwM+0ZtuBLwK/0eo3VFUBdyVZluSk1nZnVe0DSLITO Ae4sa++S4vZNy7/ywvdBS1C7/iXX+31+PNyzyLJGuA9wN3AihYkAM8AK9rySuCpod12t9ps9YPPsTnJRJKJqampuf0DJGmJ6z0skrwN+H3gw1X1zeFt7Sq5uI8VbW1qsaranxsbgwuDilJanoNiyRvYBAUv1dVf9DKz7bhJdr33lbfA6we2n1Vq81WlyTNkz5nQwW4Dni0qv7N0KYdwPSMp03A7UP1i9usqLOAF9pw1Z3A+iTL243t9a0mSZonvd3gBt4L/D3gq0keaLV/BnwMuCXJJuBJ4MK27Q7gPGASeAm4BKCr9iW5Ari3tbt8+ma3JGl+9Dkb6o+AzLJ53QztC7h0lmNtA7bNXe8kSd8Pf8EtSepkWEiSOhkWkqROhoUkqZNhIUnqZFhIkjoZFPKkToaFJKmTYSFJ6mRYSJI6GRaSpE6GhSSpk2EhSepkWEiSOhkWkqROhoUkqZNhIUnq1Oc7uLcl2ZvkoHa8Ul2Jnm8fS9v9SS5JslkkgeTnD60z8bW/vEkG2c6lySpX31eWXwaOOeg2mXArqpaC+xq6wDnAmvbZzNwLQzCBdgCnAmcAWyZDhhJ0vzpLSyq6kvAvoPKG4DtbXk7cP5Q/YYauAtYluQk4GxgZ1Xtq6r9wE5eG0CSpJ7N9z2LFVX1dFt+BljRllcCTw21291qs9Vfl8nmJBNJJqampua215K0xC3YDe6qKqDm8Hhbq2q8qsbsHxsbm6rCSJOY/LJ5tw0u0772tvgdYPdRuVavNVpckzaP5DosdwPSMp03A7UP1i9usqLOAF9pw1Z3A+iTL243t9a0mSZpHx/R14CQ3Aj8DnJhkN4NZTR8DbkmyCXgSuLA1vwM4D5gEXgIuAaiqfUmuAO5t7S6vqoNvmkuSetZbWFTVB2fZtG6GtgVcOstxtgHb5rBrkqTvk7/gliR1MiwkSZ0MC0ISJ8NCktTJsJAkdTIsJEmdDAtJUifDQpLUybCQJHUyLCRJnQwLSVInw0KS1MmwkCR1MiwkSZ0MC0ISJ8NCktTJsJAkdTpiwiLJOUkeSzKZ5LKf7o8kLSVHRFgkORr498C5wKnAB5OcurC9kqSl44gIC+AMYLKqnqiqPwduAjYscJ8kack4ZqE7MKKVwFND67uBM4cbJNkMbG6r30ry2Dz1bSk4Efizhe7EYpBPbFzoLujV/Lc5bUvm4ih/cbYNR0pYdKqqrcDWhe7H61GSiaoaX+h+SAfz3+b8OVKGofYAq4fWV7WaJGkeHClhcS+wNsnJSY4FLgJ2LHCfJGnJOCKGoarqQJJfBu4Ejga2VdXDC9ytpcThPS1W/tucJ6mqhe6DJGmRO1KGoSRJC8iwkCR1Mix0SD5mRYtRkm1J9iZ5aKH7slQYFpqVj1nR

IvZp4JyF7sRSYljoUHzMihalqvoSsG+h+7GUGBY6lJkes7JygfoiaQEZFpKkToaFDsXHrEgCDAsdmo9Zk QQYFjqEqjoATD9m5VHgFh+zosUgyY3Al4FTkuxOsmmh+/R65+M+JEmdvLKQJHUyLCRJnQwLSVIn w0KS1MmwkCR1Mi kw5DklSQPDH0O+UTeJHckWdY+v3QY5/vNJP/k8Hss/WCOiNeqSovQ/62q00Zt XFXnASRZA/wS8Kl+uiX1wysLaY4kOa69++OUtn5jkn/Ylr+e5ETgY8CPtquR32nbPpLk3iQPJvmtoeP98y T/J8kfAacswJ8kfZdXFtLheXOSB4bW/1VV3Zzkl4FPJ7kaWF5V//Gg/S4DfmL6qiTJemAtg8fBB9iR5KeB bzN4vMppDP53+hXgvv7+HOnQDAvp8Mw4DFVVO5NcwOClUe8e4Tjr2+f+tv42BuHxduC2qnoJIInP5N KCchhKmkNJjgJ+HHgJWD7KLgyuSk5rn3dW1XW9dII6DIaFNLD+jcFDF/8OcH2SNxy0/UUGVw3T7gR +IcnbAJKsTPLDwJeA85O8OcnbgZ/rv+vS7ByGkg7PwfcsPgdcD/wD4IyqeJHJl4B/AWyZblRVzyX5X0ke Av5bVX0kyY8DX04C8C3gQ1X1lSQ3A38M7GXwuHhpwfjUWUJSJ4ehJEmdDAtJUi fDQpLUybCQJHU yLCRJnQwLSVInw0KS1On/A+2Z0j36VccUAAAAAAElFTkSuQmCC\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",
"iopub.status.start": "2021-12-21T16:51:28.955883Z"
}
}
```

```
"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\nof 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": "iVBORw0KGgoAAAANSUhEUgAAAskAAAF1CAYAAAAa1Xd+AAAAAOXRFWHRTb2Z0d2FyZQB  
NYXRwbG90bGliIHZlcNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXB  
IWXMAAAsTAAALEwEAmpwYAACKM0lEQVR4nOzdd3jUVdbA8e+dSe8JKRAChA6BUENHmg0Fx  
YaKoqKubW3r6qrruq6vva26rm1ty64NO4uCgCIgtJr6IQWAiEkJCGkz9z3jztDjiGdJJNMzud58mTmV88  
MOhzunHuu0lojhBBCCCGEKGKnxdwBCCCGEEEI0N5IkCyGEEEIIUYEkyUIIYQQQ1QgSbIQQgghB  
AVSJishBBCCCFEBZIkCyGEEEIIUYEkyS2QUup7pdQNDX2sOym19imlzmmE6y5RSv3O8fhapdTC2hx  
bj/t0VERlKaWs9Y1VCCEqks/7O11XPu9Fg5IkuYk4/ody/tiVUgUuz6+ty7W01hdorf/T0Mc2R0qph5VSSyv  
ZHqmUKIZK9a3ttbTWH2utz2uguMp9yGutD2itg7TWtoa4fiX3U0qpFKXU1sa4vhCi4cjnff3I5z0opbRSqlt  
DX1fUjyTJTcTxP1SQ1joIOABC5LLtY+dxSikv90XZLH0EjFRKda6w/Wpgs9Z6ixticoxQDTQRSk1pClv  
LP9NCIE38nlfb/J5L5oVSZLdTCK1TimVqpR6SC11BPi3UipcKfWdUipDKXXc8TjO5RzXr5RmKKV+UU  
q95Dh2r1Lqgnoe21kptVQpdUIp9aNS6g2l1EdVxF2bGJ9USi13XG+hUirSZf91Sqn9SqlMpdRfqnp/tNapw  
E/AdRV2XQ/8t6Y4KsQ8Qyn1i8vzc5VS25VSOUqp1wHlsq+rUuonR3zHIFifK6XCHPs+BDoC3zpGhh5U  
SsU7RgC8HMfEKqXmKKWylFK71VK3uFz7caXU50qp/zrem2SlVFJV74HDDcD/gHmOx66vq49S6gfHv  
dKVUo84tlvVUo8opfY47rNWKdWhYqyOYyv+d7JcKfWKUioTeLy698NxTgel1NeOP4dMpdTrSikfR0yJ  
LsdFK6XylVJRnbxeITyOfN7L530tP+8rez2hjmtkON7LR5VSFse+bkqpnx2v7ZhS6jPHduX4HD+qlMpVS  
m1WdRiNF5IkNxdtgQigE3Ar5s/l347nHYEC4PVqzh8G7AAigReA95VSqh7HfgKsAtoAj3P6B5Wr2sR4D  
XAjZgTUB3gAQCMVALzluH6s436VftA5/Mc1FqVUT2CAI966vlfOa0QCXwOPYt6LPcAo10OAZx3x9  
QY6YN4TtNbXUX506IVKbjELSHWcfwXwjFJqgsrv+ix3HhAFzqotZKRXguMbHjp+rIVI+jn3BwI/AfMe9  
ugGLHKf+EZgGXAiEADcB+dW9Ly6GASIADPA01bwfyTlfQfsB+KB9sAsrXWx4zVOd7nuNGCR1jqj1  
nEI4Wnk814+72uMuRL/BEKBLsBYzD8cbnTsexJYCIRj3tt/Orafh/kWsofj3CuBzHrcu/XSWstPE/8A+4Bz  
HI/HAcWAXzXHDwCOuzxfAvzO8XgGsNtlXwCggZ1ORbzgVMKBLjs/wj4qJavqbIYH3V5/ntgvuPxY5  
gkyrkv0PEenFPFtQOAXGck4/nTwP/q+V794nh8PfCby3EK8yH3uyquewmwvrI/Q8fzeMd76YX5gLUBw  
S77nwVmOh4/Dvzosi8BKKjmvZ0OZDi7QfkAJc69k1zjavCeTuAKZVsPxVrNe/TgRr+vE+9H8AIZ3yV  
HDcM8xeMcjxfA1zZ2P+PyY/8NJcf5PNePu/r9nmvgW4Vtlkd71mCy7bbgCWOx/8F3gHiKpw3AdgJDacs  
7v5/oSX+yEhy85ChtS50PIFKBSil/uX4SiUXWAqEqapn0h5xPtBaO0cKg+p4bCyQ5bIN4GBVAAdcyxiMuj/  
NdYop1vbbW+iTV/OvWEdMXwPWOUZBrMR8K9XmvnCrGoF2fK6VilFKzlFKHHnf9CDMCURvO9/  
KEy7b9mBFWp4rvjZ+quj7xBuBzrXWp47+TrygrueiAGRWPtHX7alLuz76G96MDsF9rXVrxIlrlZjXN04  
p1Qsz0j2nnjEJ4Qnk814+76v7vK9MJODtuG5l93gQk/ivcpRz3ASgtf4JM2r9BnBUKfWOUiqliDvdt9SRJbh  
50hef3Az2BYVrrEMzXJeBSQ9UIDgMRjq/2nTpUc/yZxHjY9dqOe7ap4Zz/YL4qOhcIBr49wzgqxqAo/3qf  
wfy5JDquO73CNSv+mblKw7yXwS7bOgKHaojpNMrU200ApiuljihTx3gFcKHjK8SDmK/fKnMQ6FrJ9pO  
O365/1m0rHFPx9VX3fhwEOlbzof8fx/HXAV+6JghCtELyeS+f93V1DCjBlJmc当地+t9RGt9S1a61jMCPOBy  
tEhQ2v9mtZ6MGYEuwfwpwaMy+NJktw8BWNqrkKVUhHA3xr7hlrr/Zivwh9XZsLVCOCiRorxS2CyUm  
q0o7b2CWr+b3EZki35SslZ73omccwF+iiLnMkd/dQPIEMBvKAHKVUe07/YEmniuRUa30QWAE8q5Ty  
U0r1A27GjE7U1XWYr8ucdXkDMB90qZhSi++AdkqpPyilfJVSwUqpYY5z3wOeVEp1d0zg6KeUaqNNPf  
AhTOJtdYw6VJZMu6ru/ViF+UvoOaVUoOM1u9b7fQRcivmL57/1eA+E8GTyeX+61vp57+TjuJafUsrPse1  
z4GnHZ3wnzJyTjwCUUINV2QTG45ik3q6UGqKUGqaU8sYMjhQC9jOIq9WRJL15ehXwx/zr8TfMpKym  
cC2mvjQTeAr4DCiq4thXqWeMWutk4E7MRIzDmP+pU2s4R2MSrE6UT7TqFYfW+hgwFXgO83q7A8td  
Dvk/YBCm/ncuZtKHq2eBR5VS2UqpByq5xTRM3Voa8A3wN631j7WJrYIbgDcdIwWnfoC3gRscX/Gdi/k  
L7giwCxjvOPdlzAfrQkyN3/uY9wrgFsxfBJIAH8yHfHWqfD+06RV6EaaU4gDmz/Iql/0HgXWYD+5ldX8L  
hPBoryKf9xXPaa2f907JmH8MOH9uBO7GJLopwC+Y9/MDx/FDgJVKqTxMOdu9WusUzITtdzHv+X7M

a3/xDOJqdZyTaYQ4jTJtZLZrrRt9ZEN4NqXUB0Ca1vpRd8cihDidfN4LcToZSRanOL6a6aqUsiilJgJTgNluDku0cEqpeOAyzEi2EKIZkM97IWomq/0IV20xXzO1wXwddofWer17QxItmVLqSeA+4Fmt9V53xyOEUE+74WogZRbCCGEh3GMDP4D01/1Pa31cxX2z8DUJjp4L+utX7Pse8GzKILAE9prf/TJEELIUQzI0myEJ4EEfP2J2YCZ2pwGpgmtZ6q8sxM4AkrfVdFc6NwHQ9SMJMfWLDNZaH2+a6IUQovmQmmQhhPAsQzErraXosqXBp9Ty3POBH7TWWY7E+AdgYiPFKYQQzVqzq0mOjIzU8fHx7g5DCCHqZe3atce01lFuDKE95VdPS8UsD17R5UqpMZhR5/scrfoqO7d9JeeilLoVuBUgMDBwcK9evRogdCGEaFrVfWY3uyQ5Pj6eNWvWuDsMIYSof6XU/pqPcrtvgU+11kVKqdsWk5xNqMsFtNbVYBZ7ICkpScvnthCiJaruM1vKLYQQwrMcovySu3FUWCJXa52ptXYuHPEeMLi25wohRGshSbIQQniW1UB3pVRnxzLAV2NW4TpFKdXO5enFwDbH4wXAeUqpcKVUOHCeY5sQQRQ6za7cQgghRP1prUuVUndhklsr8IHWOlkp9QSwRms9B7hHKXUxUApkATMc52Y5eluvdlzuCa11VpO/CCGEaAYkSRZCCA+jtZ4HzKuw7TGXx38G/lzFuR8AHzRqgEK0YCUIJaSmplJYWOjuUEQd+Pn5ERcXh7e3d63PkSRZCCGEEKKWUINTCQ4OJj4+HqWUu8MRtaC1JjMzk9TUVDp37lzs86QmWQgghBCilgoLC2nTpo0kyC2IUoo2bdrUefRfkmQhhBBCiDqQBLnlqc+fmSTJQgghhBAAtRGZmJgMGDGDAgAG0bduW9u3bn3peXFxc7blr1qzhnnvuqfEeI0eObJBYlyxZwuTJkxvkWu4gNcICCCGEEC1EmzZt2LBhAwCPP/44QUFBPPDAA6f2l5aW4uVVeXqXIJREUIJSjfdYsWJFg8Ta0sIshBCCCFECzZjxgxuv/12hg0bxoMPPsiqVasYMWIEAwcOZOTIkezYsQMoP7L7+OOPc9NNNzFu3Di6dOnCa6+9dup6QUFBp44fN24cV1xxBb169eLaa69Faw3AvHnz6NWf4MHD+aee+6p04jxp59+SmJiIn379uWhhx4CwGazMWPGDPr27UtiYiKvvPIKAK+99hoJCQn069ePq6+++szfrDqQkWQhhBBCiHr4v2+T2ZqW26DXTIgN4W8X9anzeampqaxYsQKr1Upibi7Lli3Dy8uLH3/8kUceeYSvvvrqtHO2b9/O4sWLOXHiBD179uSOO+44rUXa+vXrSU5OJjY2llGjRrF8+XKSkpK47bbbWLp0KZ07d2batGm1jjMtLY2HHnqItWvXEh4eznnncfs2bPp0KEDhw4dYsuWLQBkZ2cD8Nxzz7F37158fX1PbWsMpIshGjxikpt7D56wt1hiFpKzy3kp+3p5BeXujsUITzG1KITsVqtAOTk5DB16lT69u3LfffdR3JycqXnTJo0CV9fXyIjI4mOjiY9Pf20Y4YOHUpcXBwWi4UBAwawb98+tm/fTpcuXU61U6tLkrx69WrGjRtHVFQUXI5eXHvttSxdupQuXbqQkpLC3Xffzfz58wkJCQGgX79+XHvttXz00UdVlpE0FhlJFkK0eK/8sIv3lqWw/OEJxIT4uTscUYMVe45x32cbWfzAODpHyl9DouWqz4hvYwkMDDz1+K9//Svjx4/nm2++Yd++fYwbN67Sc3x9fU89tlqlJae/g/X2hzTEMLDw9m4cSMLFizg7bff5vPPP+eDDz5g7ty5LF26lG+//Zann36azzs3N1myLCPJQohGo7Xm9Z928cfPNIBYYmuUexSW2Phs9QFK7Zp5mw83yj1Ew/JxjHYVI9rdHikQniknJ4f27dsDMHPmzAa/fs+ePUIJSWHfvn0AfPbZZ7U+d+jQofz8888cO3YMm83Gp59+ytixYzl27Bh2u53LL7+cp556inXr1mG32zl48CDjx4/n+eefJycnh7y8vAZ/PVWRf8ILIRpFUamNh77cxOwNaQBkF5Tw9vTB+Hg17L/Nv99ymOP5JQT5ejF302FuHFX71ZSEezj/G5AkWYjG8eCDD3LDDTfw1FNPMWnSpAa/vr+/P2+++SYTJ04kMDCQIUOGVHnsokWLiIuLO/X8iy++4LnnnnP8+PForZk0aRJTpkxh48aN3Hjjjdjt5nPh2WefxWazMX36dHJyctBac8899xAWFtbgr6cqyjlLsbliSkrSa9ascXcYQogzkFNQwm0fruG3lCz+dH5PQv29eXT2FiYltuO1aQOxWhquEf/lb60gM6+IywbF8fIPO/n1zxNoF+pf5fF7j53kgS828upVA+gQEdBgcTgppdZqrWvuseRB6vq5vWTHUWb8ezVf3TGSwZ3CGzEyIRretm3b6N27t7vDcLu8vDyCgoLQWnPnnXfSvXt37rvvPneHVa3K/uyq+8yu1ZCOUmqiUmqHUmq3UurhSvZ3VEotVqqtV0ptUkpd6LKvn1LqV6VUsIJqs1JKCgaF8GB5RaXc8MEq1u4/zqtXDeDO8d2YPrwTf7mwN3M3H+bBLzdhtzfMP863Hc5l7f7jXDusExf1jwVg7qbqSy5mrT7A2v3HefmHneW2v/zDTh7+ahPNbeDAE8IIs hAt37vvvsuAAQPo06cPOTk53Hbbbe4OqcHVmCQrpazAG8AFQAIwTSmVUOGwR4HPtdYDgauBNx3n egEfAbdrrfsA44CSBoteCNGsFBTbuHnmajYfyuH1awZxycD2p/bdMqYL953Tg6/WpfLYnC2VJqM/bk3no n/+Qk5+7T4mPl65Hx8vC1cMjqNzZCB9YkOYW01dstaaZsO42VRzN5wiJ3ppiPGr3syeW3RLmatPsjsDYfq+KpFXfk6k2SbJMIctFT33XcfGzZsYOvWrXz88ccEBDT8N3PuVpuR5KHAbq11ita6GJgFTKlwjAZCHI9DgTTH4/OATVrrjQBa60ytdePM3hFCuIXWmrTsAhZvP8qtH65h1b4sXr6yP+f3aXvasfec3Y3bxnTho98O8Oz328slyna75rn529l8KIdXF+087dyK8opK+WbdISb3a0d4oA8Ak/q1Y/2BbFKP51d6zsbUHFkPF/DQxF4E+Xjx94U7KCyx8cg3m+kYEcCADmE8+d02sk5Wv7SrODPeVvNXT4mMJAshmrHaJMntgYMuz1M

d21w9DkxXSqUC84C7Hdt7AFoptUAptU4p9eAZxiuEaEK/pWQy4e9LOJZXVOn+rJPFnP3yz4x87idunL  
maFXsyefbSRKYMqPgRYSilePiCXlw3vBPvLE3hg+X7Tu1buDWd3Ufz6BoVyIe/7mf30epnMC/ZcZSTx  
TauHtLx1LbJiabkoqouF3M3peFtVVyZ1IHfndWFBCnp3PPpevYeO8kzlyby/OX9OFFYwlPfba323uLM+M  
hIshCiBWioaebTgJla6zjgQuBDpZQF0z1jNHct4/elSqmzK56slLpVKbVGKbUmIyOjgUISQpyp+VuOkJJx  
ks/XHDxtn9aaB7/cRGpWAX+7KIEvh/Bur+ey9VDO1ZypTJKKf7v4j6cmxDdc99vY8shM2v5zSW76dQ  
mgE9vGY6/t5Wn5lafqC7fnUmwrxeDOoad2taxTQD94kJ5Zt52Ov95Lp3/PJdHvtmM1vpUqcVZ3aMIDfD  
mptHxhAd4s3BrOpcPimN090h6tg3m9rFd+Xr9IZbulM+ixuJjlZpkIUTzV5sk+RDQweV5nGObq5uBzwG0  
1r8CfkAkZtR5qdb6mNY6HzPKPKjiDbTW72itk7TWSVFRUXV/FUKIRrFmfxYAn6w8gK3CZLuPVx7gx  
23pPDixJzeO6syQ+AhC/b0ru8xpLBbFC5f3IyLQh3tnreeHrelsSs3hjrFdiQ7x456zu7NkRwaLdxyt8hor9hxj  
WJcIvKzlP8aeviSReyZ04+7x3bi4fyfrDzAO0tTWHcgm7ScQib3awdAsJ83f76gN73aBvOXSWWzne8c34  
0uUYF8v0V6LjcWmbgnhGgJapMkrwa6K6U6K6V8MBPz5IQ45gBwNoBSqjcmSc4AFgCJSqkAxyS+sYB  
8jylEC3CyqJRth0/QMyaY1OMF5UZWdx89wVNztzKmRxQ31bMvcXigDy9fOYCUYye565P1tA3x49JBp  
kzjhPxdI4M5J5P1vPwV5tYsftYuSQ99Xg++zPzGdk18rTrJsaF8sfzevLH83ry6lUDmJTYjufmb+fpuVvxs  
Vo4JyHm1LFXDunA/D+MIcJR0wzg523l89tG8MyifV6XaJmp0aSpdxCiDobP348CxYsKLft1Vdf5Y477qj  
ynHHjxuFs03jhhReSnZ192jGPP/44L730UrX3nj17Nlu3lqVxjz32GD/++GMdoq/ckiVLmDx58hlfp6HVmC  
RrrUuBuzAJ7zZMF4tkpdQTsqmLHYfdD9yilNoIfArM0MZx4GVMor0BWKe1ntsIr0MI0cA2HszGZtc8c  
H5PooJ9+ei3/QBk5hVx+0frCPTx4qWp/bCcQc/jUd0iuXVMF4ptdm4d0wVfL7MSm4+XhXeuG8w5CTF8  
uzGNa95byaOzt5w6b8WezFPnV0cpXytT+9G7bQjrDmQzpkcUIX41j3ZHBvmiVMP1chblyUiyEPU3bdo0  
Zs2aVW7brFmzmDZtWq3OnzdvXr0X5KiYJD/xxBOcc8459bpWS1CrmmSt9TytdQ+tdVet9dOObY9prec4  
Hm/VWo/SWvfXWg/QWi90OfcjrXUfrXVfrbVM3BOihViz/zhKwdDOEVw9pAM/7TjK1rRcrv9gFQez8n  
nj2kFEB5952/MHzuvJv2cM4foRncpt7x4TzCtXDWDNo+dyf0Iuq9as5khOIQArdh8jMsiHHjFBp18wcw+k  
rT/1NMDHi3dvSGJwp3BuGh1/xvGKMycT94SovyuuuIK5c+dSXGy68Ozbt4+0tDTOUss7rjjDpKSkujTp  
w9/+9vfKj0/Pj6eY8eOAfD000/To0cPRo8ezY4dO04d8+677zJkyBD69+/P5ZdfTn5+PitWrGDOOnDn86U9/  
YsCAAezZs4cZM2bw5ZdfAmZlvYEDB5KYmMhNN91EUVHRqfv97W9/Y9CgQSQmJrJ9+/Zav9ZPP/2  
UxMRE+vbt0MPPQSAzWZjxowZ9O3bl8TERF555RUAXnvtNRISEujXrx9XX311Hd/Vysmy1EK0ArvS  
T/DZ6oNcPbQD3aKDKz2msMSGr5fl1Ajq2v3H6REdTKi/N1cP7cgb3dz+VsrKLXbeff6JIZ3adMgsXlbLY  
zvFV3lfn+Ljd8fepge1m7899eR/On8nizfk8mIrpGVj/Yu+j9I2wB/2HRqU/swf766Y2SDxCvOnLSAEx7j+4f  
hyOaGvWbbRLjguSp3R0REMHToUL7//numTJnCrFmzuPLKK1FK8fTTTxMREYHNZuPss89m06ZN9O  
vXr9LrrF27llmzZrFhwwZKS0sZNGgQgwcPBuCyyy7jlltuAeDRRx/l/fff5+677+biyy9m8uTJXHHFFeWuV  
VhYyIwZM1i0aBE9evTg+uuv56233uIPf/gDAJGRkaxbt44333yTl156iffee6/GtyEtLY2HHnqItWvXEh4ezn  
nnncfs2bPp0KEDhw4dYssW8+2is3TkueeeY+/evfj6+lZaTlIfDdXdQgjRgLTWvLF4N6v3ZdXq+MISG4Ull  
bcgX7HnGJe9tYL3ftnLea8s5f7PN3Iwq3wf4YwTRYx+/ieen29GEux2zboDxxkcb5YMbh/mz9m9Yyi22fnn  
tEGM6111UtvGUpZgLcqmn18Gn6w6wJZDuWScKGJU1yqS9LyjkL0fik40XYyiTrwsCqVkJFmI+nItuXAtt  
fj8888ZNGgQAwcOJDk5uVxpREXLLi3j0ksvJSAggJCQEC6++OJT+7Zs2cJZZ51FYmIiH3/8McnJydXGs2  
PHDjp37kyPHj0AuOGGG1i6dOmp/ZdddhkAgwcPZt++fbV6jatXr2bcuHFERUXh5eXFtddey9KIS+nSpQs  
pKSncfffdzJ8/n5AQs0xHv379uPbaa/noo4/w8mqYMWAZSRaikWXnF+PnbcXP21rrc95csocXF+xgbI8o/n  
PT0BqPv/Hfq9l19AQvXNGPCb3MxDStNV+vO8TDX28ivk0gn/xuAHM2HuI/v+7nx23pzLv3LNqH+QPw  
94U7OJZXzAe/7OW6EZ04UVjCicJSBncMP3WP167oz+HcAnq1DSm7sdaw+BnoORHaD67166uT5K8Bi  
LEdJje/iIe/NiPEIU3aAyDf1CuTsQPikhonJnFGIFL4WC1SkyxavmpGfBvTlCITuO+++1i3bh35+fkMHjyYv  
Xv38tJLL7F69WrCw8OZMWMGhYWF9br+jBkzmD17Nv3792fmzJksWbLkjOL19fUFwGq1UlpaekbXC  
g8PZ+PGjSxYsIC3336bzz//nA8++IC5c+eydOlSvv32W55++mk2b958xsmyjCQL0Yjyiko595W1PFmHxSl+  
2JrOSwt3EOBJZeXeTIpKy0aItdanrSa3NS2XX1MyKSq1c9PMNfz56828vHAH419awv1fbCSpUwRf3jGS

xLhQ/jIpge/vPYtSm537Zm3AZtckp+Xw2ZqDp1qjvfbjLtbuPw5AUUnxZkhwa4F0+QQY4vheWvgDf/sEkzA  
2ttAi2zwWfYCz2Yia0KyY5LZe4cH86tqliCVRnknx0W8PHIxqMj9UiI8IC1FNQUBDjx4/npptuOjWKnJub  
S2BgIKGhoaSnp/P9999Xe40xY8Ywe/ZsCgoKOHHiBN9+++2pfSdOnKBdu3aUIJT<sup>W</sup>8ccfn9oeHBzMiROn  
f0vXs2dP9u3bx+7duwH48MMPGTt27Bm9xqFDh/Lzzz9z7NgxbDYbn376KWP<sup>H</sup>juXYsWPY7XYuv/xynn  
rqKdatW4fdbufgwYOMHz+e559/npycHPLyql+QqjYkSRaiEX3wy14yThSxIPkIdnvNSeTO9BP8YdZ6+sa  
G8tLU/hSW2Fm77/ip/XM2pjH6+cUs333s1LaPV+7H18vCoj+O5bYxXZi1+gCvL95N+3B/Xri8H/+5aWi5/  
sVdo4J4YkpfVu3L4s3Fu3ni262E+Xvz9CWJXDOsI1+uS+WbdYeIDPKIY0QViajTvl/M7yObTDLb0Pb8B  
EW5MORmAG7sZZKqUVWNIttUOB4vzJqPzlEND0fLxlJFuJMTJs2jY0bN55Kkvv378/AgQPp1asX11xz  
DaNGjar2/EGDBnHVVVfRv39/LrjgAoYMGXJq35NPPsmwYcMYNWoUvXr1OrX96quv5sUXX2TgwIH  
s2bPn1HY/Pz/+e9/M3XqVBITE7FYLNx+++11ej2LFi0iLi7u1M++fft47rnnGD9+PP3792fw4MF<sup>M</sup>mTKF  
Q4cOMW7cOAYMGMD06dN59tlnsdlsTJ8+ncTERAYOHMg999xT7w4erpRujNGfM5CUIKSdvfyEaMm  
y84s56/nF+HhZyDxZzNe/H8kgI/KFiopKbUx+7ReyC0qYc9coQvy8GfDEQm4e3YWHLzAfUte9v5Jlu47R  
MyaYufeMprDUzrCnf2Ri33b8/cr+A OzPPIm/t5XokKo7T2ituWfWBr7dmAbAk5f05brhncg4UcSYFxZTU  
GJjYp+2vH1dDSUUX90CKYvBLxS8/OC2ZW<sup>B</sup>p<sup>w</sup>H97f3UL7FoIt<sup>y</sup>2Ff/SjdOKLPHRgGNeN6MSADmGn  
H5+fBS84+jZ3PRuu+7rhYqklpdRarXWrqvOoz+f2iGcXMbp<sup>b</sup>JC9O7d9IUQnROLZt20bv3r1rPlA0O5X92  
VX3mS0jyUI0kn8tTSGvuJS3pg/GalH8uDW92uPfXpLCrqN5vHB5P9qF+hPo68WgjuEs22UW8UjPLWT5  
7mP07xDGjvQTfLr6ILPXH+JksY3pw8uWgu7UJ<sup>r</sup>DaBBIMTehTl/QlLtyfXm2DmTbELKoZFezLjaPiARjc  
qe<sup>q</sup>EHjDlFFuWQfxoGPMgpG+B7d9Wf05dIBTAjnnQ+yII6wjeAXhl7+XvV/avPEGGslILLz8ZSW7mfLyk  
3EII0bxJkixEIzh6opCZy/dxfc9YhnaOYEh8OIu2Vb3E8u6jebyxeDcX9Y8t1w5tTI8oktNyOZZXxLcb07Bre  
PnK/gzrHMHLC3cwc8U++sSGVJ00ViPU35vv7z2LL+8YWW5p59vGduXKpDgm929X/gRbafmOEZI74  
MRhiD8LEq+ANt1h8bOQsROO7Sr/U5hb5/jYvQiK86DvZaAURHQx96yOM0mOGwK5h6Awp/LjCo5Xv  
t0pYwcUZNc55OZCKTVRKbVDKbVbKfVwNcddrpTSSqkkx3NvpdR/IFKblVLblFJ/bqwYva0WSiRJFkI  
0Y5IkC3EGtNas3pdFxbKlf/2cQrHNzn3nmHY45/SOYUf6idNar4Fpt/bI15vx97Hy2OSEcvvO6m5qb5fvPsb  
sDYfoFxdK16ggHrsogeyCEnYfzWP68E71Xh0u2M+bIN/ys39D/b154Yr+tAv1L3/wj3+D14dC8UnzfN8y8  
7vzGLBYYdzDkLEN3hgCryeV/3nv7LoHl/w1BLSB+DHmeUQXyKplhw/2vw+W<sup>s</sup>locsoSeKEL7F1W9X  
X+OwXmN1p+2KiUUlb<sup>g</sup>DeACIAGYppRKqOS4YOBeYKXL5qmAr9Y6ERgM3KaUim+MOKW7hRCiu  
ZMkWYgzsHBrOlPf/pWfd2ac2maza/63IY3z+8QQHxkImCQZ4Mdt5UsuSmx2np+/nVX7svjLpN5EBfuW  
298nNpTwAG9mrtjHlkO5XDKg/ant1wztSESgDxf3j23Ml1hm5wI4kQarHU3g9y2DoLbQppt53vdymDYL  
Ln+/M+A6XBsJ+RIVH3tiorzYcd86H0xWB1JfJuucHyfGdGu<sup>j</sup>NJ7uSYsJJRSY<sup>e</sup>LjbNA2+Gnp<sup>y</sup>rv<sup>y</sup>FF80oy  
Qt+IS+3ibl6HAbq11ita6GJgFTKnuCeB5wHXHIEaCFRKeQH+QDFQj68BaubjZaFIkmTRQjW3+Vyizv  
X5M5MkWYgz4Owy8e3Gw6e2rTtwGN5RUzsW1auEB8ZSN<sup>e</sup>owHII<sup>f</sup>3sy8rjirRX8a2kKVyV1YOrguNO  
ub7UoRnaLZP2BbKwWxUUuCfETU/qy+IFxBPo2Qbzv3MOQuQusPrD8H6bsYq+jHtk5iq0U9LzAlF64/g  
wwM685vKH299v9A5SchD6Xlm2L6Ar2Usg5YJ7b7bD7x/KJrjNJjh0A3gGnjyQ7W8oFRsHB38ykw4qyU  
srulzK1Bw66PE91bDtFKTUI6KC1rtiS5EvgJHA<sup>y</sup>OAC8pLWu3Yo2deQj5RaihfLz8yMzM1MS5RZEa01  
mZiZ+ftXP16lIFhMR4gysTDH5w8KtRygq7Yuvl5UFW47gY7UwvmdUuWPP6R3DB8v38sPWdBYmH+  
HbTWn4eVt545pBTOrXrrLLAzCmeyRzNx1mVLfIciPNVosq19qtUTlbvZ3/DMx7AOY9CCePQuezaj63b  
T9AQdp66H5u7e635WuTyHZyaWHUxpG0ZqaY0out38CXN8GMeRDvOC4/00za8wmCqJ6njyTvXmRay  
136iXkNi5+BLuPLEn0oq3tu02KT5GoppSzAy8CMSnYPBWxALBAOLFNK/ai1TqnkOrcCtwJ07Nix4u4a  
+XhZyC8+s0UFhHCHuLg4UINTyciow7djwu38/PyIizt9MKo6kiQLUU+ZeUXsSD/B0M4RrNqbx<sup>b</sup>Kdxzi7  
dzTzk48wqlsbgv3KJ7Bn947hX0tTuOW/awj29eKifrE8cH5PYmroRDGuZzSBPlauHVb3RKT<sup>B</sup>7FsGvqGQ  
dJMp<sup>u</sup>9j4idkeX4sk2S8EirtD2oba3av4pLnHgGvKSi2gbGQ3aw9wjkmkwZRgnEqSs0wds1IQ1dv0WXaV/  
A34hUH382DMUfjuDyZx7n5O2THOuuieFltucQjo4PI8zrHNKRjoCyxx1LK3BeYopS4GrgHma61LgKNK

qeVAEnBakqy1fgd4B0wLuLoG6eNlIbtARpJFy+Pt7U3np3dHYZoAlJuIUQ9rdprRpHvP7cHof7ezN18m  
OS0XFKPfCxb9vTjk/qFM5fLuzN29MHsfrRc3hxav8aE2SAmBA/Nv7tPM7vc/o1m8y+ZdBppGOcmN  
CW0j72ieS7QaYkeTa2LkASgtMVwtXQdFmhDhzj+mWsesHsz3XJf/Lz4SACPM4uhfkHSnrZOHaUs7qD  
QOuhdCOsOSZ8iUbmSkQFAO+wbWLt/lZDXRXSnVWSvkAVwNznDu11jla60itdbzWOh74DbhYa70G  
U2IxAUApFQgMBxql1563VVFSK19XCyGaL0mShajgYFY+n646wMmi6r8K/i0lE39vK4M6hTOxT1t+2Jr  
OtxvTsKiyiXquLBbFLWO6MLFvO/y8rXWKybVFW5PLOWTqdJ2lFXGDYehMPSW8mUK1YkdaCb9n  
ThS87HJ35gkteOI8tudbeCy9sCO78FWBCjISS07xjmSDBDtaOjgrEt2tpRz1jl7+cCwW+HQWjNRzykrpSX  
XI6O1LgXuAhYA24DPtdbJSqknHKPF1XkDCFJKJWOS7X9rrTc1Rpw+XlbpkyyEaNak3EIIF+sPHOd3/1  
ID5sli/r5wB3eN78a0YR3x9To9qV25N4uk+HC8rRYm9WvHZ2sO8u/l+xgSH0GbIN9Krt5COeuRnW3VA  
C58oW7XiB1ofqdtgJ4Tqz6u6IRZYW/Q9WbUuqI2XeHwRpNIh8SZUePctLL9+ZkQ5qg0iHIspZqxDTqN  
MC3l/COg89iy4zsMc8S1HkIckyKz9tS+drqZ0lrPA+ZV2PZYFceOc3mch2kD1+ikBZwQormTkWQhHBY  
kH2Hau78R6OvFm9cOolt0EI9/u5XzX1nKugPlF5/IOlnM9iMnGN7FjFqO6NqG8ABvim32Skstmq3Fz8Ivr  
5bflp8FH0w0nSMA9i01dbwxifW/T9tEUJbKO1xk7YU3R8I/+sPrQ6C0sHxXC1cRXeH4ftizCPpcAqEdTi+  
38HeUW4TGgU8w/Pi4ufbW/0HCxeXrnGP6mric9dJFJyAvvUWPJLcU0gJOCNHcSZIsBLB4+1Fu/2gtvdq  
G8PXvR3JhYjs+vWU4/71pKCU2zdS3f+XIH3aealm1aq9pNTa8i0nIvK2WU8nxee6sHa4Luw1+exPWf1h+  
+8FVcOBX+HQa7FxY1urNcgYf75BENmj8rrkJc+ZEocOw8zCJKP+AB2GV36dN11B28BWbBLp0Pam  
HARM/+TC7LJyC6Xg3P+DHHPNtftdBSPuLn89nwAzwc8Zl7P9m4d2tmhOfKxKWsAJIZo1KbcQrU6JzY  
6XRZ1ape7oiUIe+GIjPWOC+fSW4fj7mK/5IVKM6RHF9384i8f/l8xri3bx846jvHLVAH5LycLf20pi+7BT1  
73/vJ6cl9CW9mH+ld3W/UqLTR2u0+GNph1acR6UFIK3YxKhs21aZA+YdQ3YS2D478/8/rEDTbcJrctqm  
Y/tgs2fw4g74bynar6Gc4Q3tCO0H2xKQYpyzAhwiWNNDGeSDDDkZvNTU1w755u4nO3fZCS50fl4Sbm  
FEKJ5k5Fk0arY7Jopry/ngn8sY9vhXOx2zQNfbCKvqJTXpg08lSC7CvHz5uWrBvDGNYPYn5XPha8tY87  
GNJLiw/HxKvtfKDLII/G9opvy5dTe9nnwXEdTquDkXFza281CIU5Ht0NwLMz4Dtr2Nds6jznzGGIhmlIG  
10lyPz9v+hqPvLd212jTDVCm1EIp02EDTF2ycyERZ3eLWsc1APKpmQmALb/9W4vh42WRiXtCiGZNk  
mTRqny/5TBbD+dyICufKa8v59YP17J0ZwaPTk6gR0z1Lb8m9WvHgj+MYUh8BFknixnRtU21xzcr6z8yb  
dWSvy7btneZWZUOyq9MI7HntE/zD4cbvoUbv4eYhDOPwXXyHkDGDtj8JQy9FYKiijytnMA2cMMcG  
PugeR7qSJzUl2S5Dr+uZyKa71p/xYca8owRKPytlqw2TU2u7SBE0I0T5Iki1ZDa80bi/fQJSqQn/80nnE9o/  
hxWzrn9I5hei0X6ogJ8eO/Nw3lk98N46ZRzbSZfHoyFOaUPS/MMcs8Q9kCHLYSU3fc9zKweJWWNjtk  
LHT1OmC6RXcaWTDxOWcJLflSxPHgkfAJxBG3lO363QeU9bD+NRI8qH6J8kxfcx7cHiDGUmWeuQm  
4fwWRuqShRDNldQki1ZjyY4Mth3O5cUr+hEV7Mu/rhvMqr1ZJMaFnqpPrg2lFCO7RTZipGfAbjedKTqP  
gas/Ntt2fG8muiVcAltnm7rbguOmFrnr2XBwddlIcvY+M+Ic3avhY/MJgNhBsOUR8wMw5k9mdLi+gttheiU  
fMhMRoe5Jsrc/RDsm72XugV6T6h+PqDUfR+/volJ7nfuGCyFEU5AkWbQKWMteX7yb9mH+XDLQjD4q  
pRjWpQWVTNRG7iEzGW/7d6asIXaA6Skc2sFMjNs62zxXji+R4kebhPjIZvPcmSxHN0B5RWwun13Wj  
UJZnnzU1svHrMSXm2pW0YO61ySDKbnY8rX5h4OMJDcJGUkWQjR3Um4hPF6pzc4PW9NZu/84t43tgr  
c7V6+ri42flSWvteWceAamtVrBcbPSXMIUs8hGh2EmSd63zCy2ERRtSiuy9kJxPhzdas6N6tlwr8OVb7BJy  
qN7QVSPyhcMqauQ9o6Je1ngHWhGhusqdqBJkEE6WzQR50iydLgQQjRXMpIsPEpmXhH3f7GR/CLz1X  
teUSm7M/IoLrUTHezLIUkd3BxhLeUdhd3m9Xhrp9d+/OcLcwG3whr/w0/PW1auPW5zGzvcxnMfwgyv  
GHwDLMtuheg4dhOyNhuRp19q5/E2KyEtjd11IHRdS+1cGo3oOyxjCQ3CedIsiTJQojmqoUMqQlRO//bk  
MaSHRkoBVaLIjrElxtGdOKlqf359u7RLaf2cdsc05pt71I4eaz252WlmJZq5/6f6U6x+l1HT+FBZn/CxYAy  
XPns8w25yS9jO2m3CKqEeqRG1NIXNnEvfqUWoBj8p43oCC8mU7I9DCnkmQptxBCNFMykiyazM70E/  
x94Q4evqA3nSMDG+UeC5KP0DMmmM9uG9Eo128yW74xSW7Bcdj2LSTdWLvzMveYHr9+oTDyblj0  
RFIPYYCQWOg4Ag6sgE6jzbY2XU2CmL7FjCZ3HdcYr6jxhMSaUonjeyGsd1KTuPlaxLl/MyyRVVEo/K  
WcgshRDMnI8miyXyy8gALktO57M3lrN2fVa9rzF5/iPs+20BOQclp+zLzili9L4vz+7aQZaGrcuII7F8OQ28

zi2e49jauSdaesoUwht4GSTebPsSuJvwFzn6srKuE1Rsiuzu6YBSVjSy3FM5eyZl76l9uAab38riHGyYmUS  
MZSRZCNHeSJIsmss2xXBv3iQgkL8GHauvy5fvPhmk+q4J8/7eKb9YeY+vYKDmUXINv347Z07BrO7xPT  
UCG7x9Y5gIY+l5qffb+YGuWa2G1wfF9ZTa1vEEEx+2UzYcxU/Gs66v/y2qF6Quds8boz2b40pJM7xQJ9Zk  
txrEgyc3iAhiZr5ykiyEKKZkyRZNIIID2QXsyTjJxf1j+eqOkfSNDeHuT9fz657MU8dknCjiqn/9yshnFzHy2U  
WMf2kJ046cOLV/99ET7Mk4yaUD23M4p5BL31hOclrZohnztxyhQ4Q/Ce1CmvS1Nbkr00LtuheZqKdtps  
a5ZrkHDT9kOvTnSHaZfS4pdUkO0eS4cySZNGkvKUFnBCimZMkWTSJX3ZIADCmRxQRgT7MvGkon  
doE8PuP13IwK5/jJ4uZ/t5KNqXmMLJbJKO6RXIk5D3f0k5dY0FyekAPDSxF1/dMRIvi+KGD1aTll3AicI  
Slu/O5PyEtnVaGKTZyU0zK+H1udQ8j+4NkT0heXbN5zo7W9SnO4MzMQ7rZFbBa0mC2pb1fa7vxD3R5  
KQFnBCiuZMkuZXRWvP95sPkF5c2yPU+XXWAS99czqbU7GqPW7rrGDEhvnsPDgIgxM+b924Ygs2uu  
eW/a7jh36vYm3mS925I4qWp/Xlxan8uGdieORvTTtUfz99yhAEdwmgb6kePmGD+e/NQCkts3PrhGr7fc0R  
im52JLb0e2ZkMO5NkpcpKLrbOgV0/mhXyKpPl+AfFmYwkR7ewemQAq5dJIEFGklsQaQEnhGjuJEluZd  
YdyOaOj9fx3rK9Z3wtu13z+k+7WX8gm8veXME/F+2itJKvTm12zfLdxxjdLarcKG/nyEBev2YQO9NPsd  
Utl7enD2KUy3LP1w7rSGGJna/XpXIou4DNh3I4v09ZEtwtOph/XD2A5LRcHv1mC5FBvgzqGH7Gr8utdn  
4P0X3MRDqnvpeb359fBx9fDu+fA3t+Ov3czD1mMY3gevxDIbwz+EdA+6T6xe1uzpILSZJbjFPdLaTcQgj  
RTEkLuFbmx22mZGH2hkPcPaFbtaUJRau2rnjrV6wWxcX9Y5ncrx3RIWXtsX5NyeRQdgFPTunDyr1Z/P  
2Hnfzn1330ahtCz7bBTB/eic6RgSSn5ZCdX8KYHpGn3WNMjyjeuyEJP28rI7uW39+3fSgDOoTx8coD2L  
XZVnFS3tm9Y/jT+T15Yf4Ozk2IwWJpwaUWdrtZSjpxavntUT3grtVQkA1o+PImWPwMdBlf1toNyjb1Kf  
cxOoFd64Ev7D6x+9Ole2B1ZIktyC+MpIshGjmZCS5IVm0LR0fq4WUjJNsPpRT7bGfrDzA5kM5nCgs4Yn  
vtjLiuZ9YmHzk1P7P1xwkhM+LqUkdeP2aQbw9fRDjekaTU1DCh7/t58p//creYydZtsshuE6SuxqQq+Y0xJ  
kp+nDO7H7aB5vLdlNz5hgukQFnXbMHW078upVA7j/vB61fRuap+N7oSgXYgecv+yO3QYAh2Gms4U  
qavNctOuMvdAmy71v39QNHj51P98dwp1dLiQJLnFkBZwQojmTpLkVuRgVj470/O4fVxXfKwWZq9Pq/  
LY/OJS3li8m+FdIvjxj2P58Y9j6RkTzF9mbyEnv4Sc/BK+33KESwa2P7WK3cS+7U6tbDf37tHY7Jpr3/2Nb  
zem0Sc2hMgg3zrHPLlfO0L9vTmWV1xlazelFJcMbF+v6zcraevN79iB1R834Fqzit7ip0E7httpZC9v371yJ  
4gfjS06w8Blf9jSzQ/MnFPCNHC SZLswbTW2Jx1CpSVWIw2sD0TekUzZ2NapTXEADNX7ONYXjF/Or8  
nSim6RQfxwhX9yDpZzHPztzFnUxrFpXamDu5Q6fndY4L5701DySsqZfuRE5zVPaper8HP28oVg80o4X19  
WvikvJqkrTdLStfUgs3LB8b+CdLWwc4FZlv2frCX1q+zhSfoeQHcttSUjYgWQVrACSGaO0mSPdTm1Bz  
Oe2UpI721gqJSGwCLth2la1Qg8ZGBXDKwPcfyilju0qfYKaeghLeX7GFCr2gGdyprqdW3fSg3j+7Mp6sO8  
vpPu+jVNpi+7avuSdy3fSgzbxpK3/YhXDIwtt6v5Q/ndOft6YPp2z603tdoEdLWQ9tEswJeTfpPg/B4+O1JK  
D55Zp0thHADGUkWQjR3kiR7mFKbnnd/2sWlby4n62QxGw9m89KCHZwoLGH13kzO6W1KFsb3iiLEz4  
v/rT902jVe/2kXuYWlldb43ndODzpE+JOeW8SVSR1q7Ek8qGM43919Fr3a1n+Bj2A/75bf2q0mdjsc3gjtBt  
TueKs3nP8MHN0Kn1wFRzaZ7a11JFm0ON5W89khSbIQormSJNnDvL54Ny8t3MnEvm1ZdP9YrhveiXe  
X7eWZedspesWnOdiTJvl5WJvVrx/zkI2w7nAuYlm5Pz93Ku8v2clVSB/rEnj5y6+9j5cUr+jOwYxiXDmx/2n  
5RT5m7oTiv5npkV70mwaXvwP7l8NPT4BMEgfUraxGiqSml8LFaKLbpmg8WQgg3kAK+Fiw7v5hQf+9T  
o7k2u+bTVQcY1zOK168ZBMBfJvXmt5RMPI11gPAAbwZ1DDt1/vUj4pm76TAXvraMi/rFULxqZ37yEW  
aMjOevkxOqvO/wLm345vejGvW1tTq1nbRXUb+pYLHAV7eYUeSWvNqgaHV8vCwykiyEaLZkJLmZO  
ZJTWKvj/r18LwOf/IF/L993atvSXRmk5xZx9ZCyyXR+3lZemzYQH6uFCb1i8LKW/ZH3bhfcsgcncMfYrv  
ywNZ0FW4/w18kJPH5xH6wtud9wS5S2HrwDILiebez6Xg7XfQMTn2/4uIRoRD5eFoptNneHIYQQIZKR5  
GbKx63p/O6/a5h54xDG9Yyu9Bi7XfP0vG28/8te/L2tvPbTLqYmxRHs580Xaw4SEejDhF7IW6X1bhfcde  
MJjr49BZpoQHePDixFzeO6kx6bqHnT45rrtLWQ9t+9e/O0GVsw8YjRBpwtoZSRZCNFsyktyEcgpKOJZ  
XVOX+T1YdAODFBTuw20+v08vOL+a2j9by/i97mTEynk9vHU52fgnv/7KXrJPF/LA1nUsGtD/VpN9Vj5  
hgwgKqXigiKthXEmR3sdvMxLvKFhERwoP5eFkokZpkIUQzJULyE5m/5QjjXlzMIff/6Fa1P/0shPbeQJTuo  
0jMmmOS0XOa7rGwHsGxXBue/upTF24/ymKMkYkCHMCb2act7y/Yyc8U+SmyaK4fENdVLEg3l2E4oy

a97PbIQVVBKTVRK7VBK7VZKPVzNcZcrpbRSKsllWz+l1K9KqWSl1GallF9V558pH6vUJAshmi9Jkht  
ZfnEpf/piI7d/tBaAlIyTbHV0k3D19bpD2DW8ce0gukcH8feFOyi12SkssfH4nGSue38VwX7ezL5zFDeN7nz  
qvD+e14OTxaW8tmgX/eJCz6jVmnnCTQ+vMb0mSRQNQSlmBN4ALgARgmlLqtJm4Sqlg4F5gpcs2L+Aj4  
HatdR9gHFDSWLH6eFkpkRZCNFMSZLcyP71cwpfrkvrlrvHd+P7eMVgULEhOL3eM1pov1hxkSHw43a  
KDUp+8HuzJOMILC3cy+Z+/MHPFPmaMjOe7u0efVhLRIyaYSwaYVmxTB8socotzbBcsegKY6FNN3d  
HIZzDUGC31jpFa10MzAKmVHLck8DzgOts4fOATVrrjQBa60ytdaPNrPOxKllxTwjRbEmS3MgWJB9hS  
HwED5zf7ahfgYJj2DBlvKIFGv3Hyfl2EmmJpmuFOf3aUti+1De/nkPJwpL+O9NQ3n84j74eVsrxveDE3syf  
XhHLh0kSXKLkrEDZk4CbYPrvgZL5X++QtRRe+Cgy/NUx7ZTlFKDgA5a67kVzu0BaKXUAqXUOqXU  
g1XdRCI1q1JqjVJqTUZGRr0ClRZwQojmTLpbNKLU4/lsp3KCRy7sdWrB+X3a8sR3W9l77CSdIwMB+G  
JNKgE+ViYltgNMk/3nLk9kzsY07hbtdoJdwDtQv156pLExnshomp5R2Hen6DUMRgX1tG0YrO4/Ptz6Yv  
Qtj/0OK9sW/YBmDnZPL7hO4gu+29EiMaklLIALwMzKntBYwGhgD5wCKI1Fqt9aKKB2qt3wHeAUhK  
SqrX7DsfLwuFJZIkCyGaJxlJbkQ/bT8KcGopaIDzHcsrL3BMzDuSU8h3m9KY3K8dgb5l/2bpExvKny/oX  
WOCLNxss/ewdTbkpMLxfbDqHdj2v7L9B1fBT0/Bt/dAicu32oufgaJcmCEJsmhwh4AOLs/jHNucgoG+wB  
KI1D5gODDHMXkvFViqtT6mtc4H5gGDGitQb6tFyi2EEM2WJMmN6MdtR+kSGUiXqKBT29qH+ZPYP  
pQFyUcoLLFx24drALh1TBd3hSnOxOEN4BsKt/8Cd6wwi4EseR7sjr/4lzwLXv5w4jCs+4/Zdmw3bPoMhv  
wOonq6LXThsVYD3ZVSnZVSPsDVwBznTq11jtY6Umsdr7WOB34DLtZarwEWAIKqQDHJL6xwNbG  
ClS6WwghmjNJkhtJXIEpv+3J5Ozepy8KMrFvW9YfyObOj9exMTWHV64aQLfoYDdEKc5Y2nqI7W+Wg  
7ZYyexDkLENtn4DB36DPT/B+D9Dp1Gw7GUoKYClL4DVF0bd6+7ohQfSWpcCd2ES3m3A51rrZKXU  
E0qpi2s49zimFGM1sAFYV0ndcoORmmQhRHMMNmN5JddGRTb7JzdO+a0fef3ieHFBTtYtP0o95/bg/P  
6tHVDhOKMIRbBkS0w4vdI2/pcCktfgiXPQXBbCIwyI8btB5tJegsegc1fwIi7IKjyVRWFOfNa63mYUgnX  
bY9Vcey4Cs8/wrSBazx2OxQcx9eipQWcEKLZkpHkRvLD1qOE+nuT1Cn8tH3dooMZ1a0NVybFcdeED2  
37VZgDtlJ3R9G4jm4Fe0n5/sYWK4x72CwQsnepjL4PfAlhfjTEnwVrPjDlFzKKLFqzjZ/Ci12I0RISkyyEaL  
YkSW4ENrtm8Y6jjOsZhZe18rf4o5uH8cIV/VFKNXF0TcBWCv8cDCv+4e5IGfaevO74iIgvS+GmL4QFA  
NIN5VtH/8X83vYrRAY2TQxCtEcOb5FCdfZFEuSLIRopqTcohH8lpJJ1sniSkstnDwyOXbK2A4nM2DvM  
jjrfndH03jS1oNfGIR1Kr/dYoFr4TSAvD2L9veaQT8bhG07dekYQrR7DiS5FDbcYpLw9wbixBCVEGS5A  
amtebVH3cSHezLeQIVJ8kezTnCmrYetDaT2jxR2gYzilzZ6wtpV/k5cUmNGpIQLUKgi0m2H6fE1qmGg4  
UQwj1qVW6hlJqolNqhlNqtlHq4kv0dlVKLIVLrIVKblFIXVrI/Tyn1QEMF3lwt2ZnB6n3Hufvs7IWukOfxn  
ElyYTZk73drKI2mpNDUJFcstRBC1CwwElCElB6nxKax2+u1FokQQjSqGpNkpZQVeAO4AEgApimlEio  
c9iimzdBATE/ONyvsfxn4/szDbd601vx94Q7iwv25KqlDzSd4qsMbIKCNeexMmD1NejLYSyVJFqI+rN4Q  
EEFwaSaA1CULIZql2owkDwV2a61TtNbFwCxgSoVjNBDieBwKpDl3KKUuAfYCyWccbTM3f8sRthzK5  
b5zeuDj1UhzIlf8E776nf55nbISqndeavfh9S1jROTq9Ji0xYtcSpYfTw3SU5bZ37HDnBrGEK0WIHRBJYe  
B5AOF0KIZqk2mVx74KDL81THNlePA9OVUqmY3px3AyilgoCHgP+r7gZKqVuVUmuUUmsyMjJqGXr  
zYrNr/v7DTrpFB3HJwIpvTwNJ2wALHzUT4g6thU2fw29v1yK4Uvj+QVj5VuPE5SpjG9iKIG4IxPTx4CR  
5gxktD23F3xgIcSaCogksdowkS69kIUQz1FDDndOAmVrrOOBC4EOllAWTPL+itc6r7mSt9Tta6yStdVJU  
VFQDhdS0vt9ymN1H8/jDOd2xWhppotqS50w3hbtWwT3rodeFsHU22G3Vn5dzwJQGZO5pnLhcubZFix0  
IaRvN5D1Pk7a+6kl7QoiaBUXjXyLIFkKI5qs23S0OAa7DZXGOba5uBiYCaK1/VUr5AZHAMOAKpdQL  
QBhgV0oVaq1fP9PAmxOtNW8s3kOXqEAu6FtFV4MzdWgt7PweJjwKfqFmW59LYdu3cOBXs1hFVTIdJ  
RlZexq/20TaBvANhYguJolc84EpCWnTtfHu2Zh+eRVWvXv69txD5h8pQoj6CYrBv0hGkoUQzVdtkuTV  
QHeIVGdMcnw1cE2FYw4AZwMzlVK9AT8gQ2t9lvMApdTjQJ6nJcgAi3ccZdvhXF68ol/jjiL7h8Ow28u2  
dT/frN6W/E31SXKWYwS5MAfysyCwTePECI4R1gEmEW83oGxbS0ySbaXw6+umrKJ9hdZtFisMuNY9c  
QnhCQKj8LIVEECh1CQLIZqlGpNkrXWpUuouYAFgBT7QWicrpZ4A1mit5wD3A+8qpe7DTOKbobUnf  
sfuYLeb5Yi9fNFa8/pPu2kf5s8lfSNqP1JbWmxmeNfm2IOrYddCOPtv4Btctt03CHqcD1v/BxOfB2sVf5yuZ

RZZeypPkrU2o6POP7agGPDyqTk2V6VFpuvDiDvN8+jeYPU1SXLiFXW7VIxNsNhNjVa+1Ie1fbhZFmfQyJFzc+PcTojUJMn3kI1UORTKSLIRohmpV6y1nqe17qG17qq1ftqx7TFHgozWeqvWepTWur/WeoDWemE13hca/1Sw4bvJvMfhfPAIsJK/dmse5ANvcNC8T7IV6w4eOaz9ca3p0AMydB0Ymaj1/5FvhHwNBbT9/X51KTyO1fXvX5WXvA19F8pKq65KUvwit94NW+5ufz62uOq6L0ZPOPB2dbNKs3tE2Ewxvrfq2qLPrie/o9oa7ZIWSvwbvQOh+buPfS4jWJsjMP4kiW80thBDNUiP1KfNwB36FYzsoWvsxL8zfTmSQD5ecmAVFubB3ac3np62D9M0msf3ocijMrf749K3QcYQZOa6o+3ngHWBKlqqSuQc6jwFlqbpl3M75ENUbLn4duk4wsdnr+BfX4Q3mt2tbtNiBpk65rteqTGmx+UfllicPmHxjpW8/8mlWxlcLWOdDzgvJLSwshGoZj1b1IIUOJzXO/eBRCTFySJNeV3QbHdgJwfP4zJB88xjMTwvDa8KHZX5uWZ1u+Bos3XPxPMyHvo8uqTpTtdji+F9p0qXy/TwD0mAjb5pjEriJbCWQfgKhepl1ZViUjyYW5JpHtNQkGXQd9LjMJf216MJ88Bpu+MO3otn1r6qbDXJaZjR0AxScqv29dpSwxq/id9zRYvOA/k6tPlIvz4eCq07dn7ITcw9Xfa99SKMgyI/VCiIbnKLeIUjkykiyEaJYkSa6r4/ugtJDFlhG0tacze9Q+zsv8yNQWD7oBju2qfmRYa0ieDd3OhkHXw9SzLoa1rxf+fG5h6C0ECKqmjfW7yrIz4Tv/nD6iO3x/aBtZuJcm66V11sc+M0c09kxz9JZLIGbhP/Hx+Hr38Hxt8Cen6DTqPJ11rGDzO/U1TVfqybJ35jOGUNvhRvnmUR5XjUrnf/yCrx/bvlFVApz4P1zYOFfqr/Xlq/BJxi6nPmcQshThfQBo0ySbKthjaWQgjhBpIk11WGqYX9QE/mZPRAem9/E9Z/ZBLeXpMBDUc2VX1+6hrITS0boex9EUQnVF2m4RyBra47RI/zYcyDsP5DmHNX+b7JzvMjupqfrJTT+xbvW2pWx+swzDyP6gVefrVLkvf+DN3OhbvXmZ8rPii/P6qXqafe90vN16pOaRFsnwu9J5sJhW26QtLNsH9F5aPCWsOWr8zjJc+Wbf/tLZMoZ+6u+l62EjMq3utC8PY7s7iFEJWzemHzb+OoSZZyCyFE8yNJch3Z07cB0DVhMIHn/RXyjpha39F/LKvFTdtQdkL2AVjxukm8wIyGWn1MratT/GgzmltafPoNM12S3KooBRP+AuP+bGp2v73n9POdi8lFuaZEwtXeZWaFPGftrdXLMeHO5XXYSmDpS+XPPb7fvL7u55Zd38u3/LUtFogfZe7h6tBaWPdh+W1amxUEM3ac/hr3LIainPLID30uBbTp71HRkc3mHwgxibD7B1N2UXAcfn3D7M8+UP74ojxY8Bf47j746mZT1iGlFkI0Kpt/JJEqRxYTEUI0S5Ik11Hugc2k6kgGdu9oJrj1vcKM4oa2h6BoCIkrPwK75Dnz1f4XM8xoPI3ZuTVuSAIQPxZUJJvJvRVIJVieiEH12KRknEPw4i7zMj2McdIadYeU6IQ0KYs0XatDy7INiPfFfssxw40XSmco9I7F8BPT8Kqd8qOcY4Ox59FteLHmFX/ju8r27bwMTPqfWRL2baUJTD/IfOeVZT8tVltsMu4sm1RPSCmb+WTfpO/AWWFaZ9AQKQZTf71TfOPhMQrTcLsWhaTstj0RE7+xoxOxw40f75CiEZjD4yWmmQhRLMISXId2dK3ssvenmGd25gR3Cveh7F/KjsgdkBZklxaDNu+g/B42P4dvHcOnEg7fYQyfjSgTh9tBTMSHNHFjMjWxvA7zO+t35Sd36aLidVZsuFal3zgV9D20xPd2IFQnFdWluBMRJO/KSvX2LfMJN9RvaqPyVnr7Hx9J46Utaz72ZEQa11WFrFzvpl051RSCNvnmdiUq3f5a/e5BA7+Bjkui0BqbZLqLmMhrCOMutfUSy//ByRMKRVfdx1Ndr4n926EP+2GW5ecPiouhGhQOjCaSCRJFkI0T5Ik14WtlJC8vaT7daZtaBW1qrEDzUhtQbYZnSzKgQteND9HNpnFNxpOLH9OQIQZE1XSV1y1p6qO1tUJjTO1BYnzy473zmCHNbRjK66jiTvXWZiihty+usAk/CXFMC0701t8bGdcHSrSUT3LjMJfk0JfFQvM5rrHHneOgfQpovGtm/NiPWeRXBwpdlWkm8WT3HatdB0yKis/KHPZY5rupRcHN5gRq2d+4b8zrSbshXD2Ich3NF9wzVJztpjYnQd4RdCNC0VFG1awJXKxD0hRPMjSXId2DJT8KYEa0zvqg9y1iUf3mg6JDhLBIBdaia1Tfp7+VXznDqfZepmS4vKtltJtmrrh65Mn0ug/QtppQhJ7VsBNnqbRJE15Hkfcugw9DTJ6hF9jD9l9PWmyS15CRc+KKpv97ytYkrN7XmUgswo9jxo829nKO80Qlw0asmKV3yHCx+1rSou+RNk9Amf23O1Rp+edm0les85vRrt+kKbfuVHQ+OFntepqUdmDZ5U16Hic9BTEJZi7pyI8kpLXPpbCFaMBUcj8qRtdmUSUhhGhikiTXwaGdpmY4qsuAqg9q5xiBPbgSdswr68YA0Pdy04e4MvGjTau31DVI23IOmtHPuiZvCRcDyrRA0/bySXZE17KR5PwsM8GtskTXYjXJZ9p6U2IREAkJ15hjk78xCS9UnrhWpvNZpp3d/uWmxKPPpSZBHnGXeZ8OrYExD5jJgwITYOdCM5lu53wTw5g/nV5q4dTnUtNi7ug2U2ucPBu6jDcj9E49zofht5vHAW3MPwCy95ftdx1xF0I0CUuw6ZXsVXCshiOFEKLpSZJcB0dTNgDQKzGp6oMC25iyhpVvm0lite2Q0GkkoMqST6hdZ4vKhMSaFfqco6uuSXabrmUVGvHJDxdVjNcUexAOLzJTNpLmGK6XvS51CSUK/9IRnwje9QupnhHMv39Q+a3830ZdrsZbQ/rCAOUldtXWmAS5MXPQHhn6H911dd2XuvN4fB8vJkkWN37rpQZTXaOJBefNKv41aWsRQhxxyC2wLgU5Dh5kiEEOJ0Xu4OoCWxp2/jsIqmXVRk9QfGDjQ1sv4R0Hls7S7uHw7t+pk633EPm23OFe/qUwbQ9zI4sMI8jnBJ/iK6mtKJ+Q+bRL7vFdBheNWvY+Vb5v

Gpvs4Xw9z7TTlHn8vKLxxSncjuZoWt9C2mLVtkd7PdLwSu+9p08HCOFHccAUFlYeFfzUTHS96qehQZIKlzXPURZB80z739IHf9fGEdTQt7KDsfaRZCGalCYYLE3tUyQjyUKI5kdGkiuw2TWvLdrF/C1H0C6LbtjsmvCTe8gO6lbzRZyT3irrxlCd+LMgdZWZKAadmJNkn6NTyrXXS21Fy4R9evuzAOVq68m2zUt9l71Q98c75OoJiHCPdmJHyLo7Ev6oR6Mo465LBdKRw1X6wqRV2sljMMSfSTOKaeGXN1+99EYz4vflJuqmsxKUqYR3LRpJde0kLIZqO47PNV5JkIUQzJElyBW//vIeXf9jJ7R+tZcoby1m0LZ1d6Sf4KTmVTjqt+kl7Tp1Gmd/9rqrzbuMMzXIO+aZ51l7zChpbUdrXQXHmKWv2/Uvvz06wUy+6z/NjNBarFVfo003U1KROLX8cf2uNhPjouoyvW0zdzwOLtxnlrokzMR7/iCnzaGjhnUznkYJsl1UJpdxCiCYVEIENhX9RprsjEUKI00i5hYuNB7N55YedTEpsx7ieUbz64y5u/o+ZSNdVHeJcXxvRXQfUfKEOQ+H+HeCot6u1rhMgsif8/IKZJJe5x6x8V19T/wNUWO41JBbu22piqyn5tljg97+d3o2j35Vmwl5ILRY4KXfeVeYfArV5X+IGwx+31/0etRXW0fzO3m9qtINiKu86IoRoPBYrxwkloFiSZCFE8yNJssPJolLunbWemBA/nrk0kdAAby4eEMuyncollXR7tBRWAhlnfrV7oJ1TZDBjNaOewi+vAk2f2kSuIqlCXXhG1T59roknoFtTt+mVP2SV6Xq9r40VoIMLknyAelsITyOUmoi8A/ACrynta5kGUtQSI0OfAkM0VqvcdneEdgKPK61fqkxYz2uwggoyWrMWwghRL1Ikuzw9LxtHMjKZ9atIwgNMHXEv15Wzklw1ANvWwS+IRDVs3EDSbgUol6EBY+AvVSSt8bi7JV8fL8Zse9xnnvjEaKBKKWswBvAuUAqsFopNUdrvbxCCcHAvcDKSi7zMvB9Y8cKcNwSTnSpJMIciOZHapKBUpud/60/xBWD4xjaOeL0Aw5vMivDDb/D9PFtTBaL6W6R75jIIInWyjcM/HHyCIT0ZTh6Vf4wITzIU2K21TtFaFwOzgCmVHPck8DxQ6LpRKXUJsBdIbuQ4AcixhBMsSbIQohmSJBNYkpLyWIbY3tEV37AkufANxSG/75pAup9sVmmGqTjQmNRypRcpCwxz+V9Fp6jPXQ5XmqY9spSqlBQAet9dwK24OAh4D/q+kmSqlbIVJrlFJrMjLq3+c40yuasNJjZvEgIYRoRiRJBn5LMZNGhnWpZBQ5bT3smAsj7gT/sKYJyGKBya/A8DshMKpp7tkahXcybeZARpJFq6GUsmDKKe6vZPfjwCta6xozVq31O1rrJK11UIRU/T+ndvsmYMUOh9bW+xpCCNEYpCYZkyR3iw4iMsj39J1LnjMrwjmXNG4qHYaaH9F4nJP3QMpaHCc5BHRweR7n2OYUDPQFlijT4aYtMEcpdTEwDLhCKfUCEAbYIVKFuvXGyvYvX59zIODq8p6sAshRDPQ6pPkUpud1XuzuHRQ+9N3Ht1ulkae8Cj4hTZ9cKJxOZPk4FjwCXBvLEI0nNVAd6VUZ0xyfDVwjXOn1joHOLVsqFJqCfCAo7vFWS7bHwfYgjNBBij1CWG/tROdDv7WmLcRQog6a/XIfs565OFdKml15lzWuaYljkXL5OxwIfXIwoNorUuBu4AFwDbgc611slLqCcdochPi42Uh2doLDq4Gu93d4QghxCmtfiR5pbMeuXMISXLaetMFwZlMCc/iHEmWUgvhYbTW84B5FbY9VsWx46rY/niDB1YJH6uFTZZeXF4ADK2l1+iXggh3KjVjyT/lpJJ16hAooIrqUdO2wDtBtRvWWjR/IXHg5ffma1qKIQ4Iz5eFjbg6D8vJRdCiGakVSfJpTY7q/cdr7zUoqQQjm6F2IFNH5hoGn4hcNdqGDzD3ZEI0Wr5WC3stUWbTj4HV7k7HGEOKVVl1skp+WSV1RaeZKcnmxWvJmk2bO5drgQQjQ5by9FsU1Dh2FwQEaShRDNR6seSa6+P/I681uSZCGEaDSBPl6cLLahOwyD43sh76i7QxJCCCVJ8lr9x+nc2Qg0cF+p+9M2wABbSA0rsnjEkKI1iI2zJ/iUjvHIx0DEgdXujcgIYRwaNVJ8pZDOSS2DzVth35+EbZ8Vbbz8AYziyyT9oQQotF0iPAHYJ93d7D6SsmFEKLZaLVJcmZeEWk5hfSNDYJv74HFT8G3f4CC41CcD0e3SamFEEI0srhws5DPwVwbxA2BHd+D3ebmqIQQohUnyVvScrFg55IDz8L6D6H/NVCUC7++AelbQNskSRZCiEYWF25GklOPF8CQmyFrD2yf6+aoBCiFXe3SE7N4kXvfxG9ZxmM+zOMexhKTsJvb4GymoPaDXBrjEII4ekCfLyIDPLhYFY+j0C4Z3hl1eg90VS7iaEcKvWOZJsK2Xwuke43LoMxv/FJMgAYx+G4pPwy8sQGA0hse6NUwghWoH24QECPJ4PFiuMusd0F9q3zN1hCSFaudaXJNtK4ZvbGJb3I3Pa3AxjHyzbF5MAfS8DW7FM2hNCiCbSIDzfFuAKX0LjDajyUII4UatL0ne+Als+ZLnSq7mUOKdp+8f+xAoC3QY0vSxCSFEK9QhIoC07AJsdg3efjD8DtjzExze6O7QhBCtWOtLkg+spNivDW/bLjLt3yqK6gl3rIDh1STQQgghGlxcuD8INs2R3EKzYcjN4BsK8x8xLTqFEMINWl+SnLaewwG9AUWf2JDKj4nuDT4BTRqWEEK0Vh0cbeBSs/LNBr9QmPgM7P8FVv3LjZEJIVqz1pUkF+dDxjaS6UxcuD/hgT7ujkgIIVq9DhGOXsnOumSAAddCj4nw4+NwbJd7AhNCtGqtK0k+shm0nWUnO9A3tpJSCyGEEE0uNswPpTBt4JyUgov+Ad7+8M3tZtK1EEI0odaVJB/eAMCinPYkxkmSLIQQzYGvl5WYYL+yDhdOwW3hwpg0Boz6VoIIzPQ60qS09ZT7B/NUcLpW9mkPSGEEG7R1cLf9EquqO/ENMXfnbtG76wIQQrVarS5LTAnoB0LeqSXtCCCGaXFx4QNnEPVdKwbDb4GiyLDAihGhSrSdJLsqDjB1stnemU5sA2gT5ujsiIYQQDh3C/TmSW0hxaSUt3xKngn+EGU0WQogm0nqS5COb

AM1Pue0Z2CHM3dEIIYRwERcRgF3D4ZyC03d6+0PSjbBjHmTtbfrghBCtUutJktM2APDLyTgGdx3byx  
CCCHKiQv3Bzh98p7TkN+BxQqr3m3CqIQQrVkrSpLXU+DflgzCGCRJshBCNCvOBUUOVlaXDBASC  
wlTYP2HcCK9CSMTQrRWrSpJ3u/TAz9vC73aBbs7GiGEEC7ahfptajKO1w4nfUA2Evhk6lQdKLphgNC  
tEqI0kuzIXMXawt7US/9mF4W1vHyxZCiJbCy2qhXWglvZJdxSTA1JlwZAt8fgPYSpOsPiFE69M6ssWM  
7QD8nNOWgR3D3BuLEEKISnWODGRXel71B/U4Hya/AnsWwdw/Nk1gQohWqXUkyblpABywtZFJe0I  
IOUwN6BDGjvQT5BfXsAT14Btg+J2w7r+QuadphNCtDqtI0nOM5M80nUYg2QkWQghmqUBHcKw2T  
WbU3NqPnjk3aCssHZmo8clhGidWkeSfOIwpXgREBpNdIifu6MRQohGpZSaqJTaoZTarZR6uJrjLldKaaV  
UkuP5uUqptUqpzY7fE5ouapMkA2w4mF3zwSHtoNckWP8RIBQ2alxCiNaplSTJRzhGGAM7SamFEMKz  
KaWswBvABUACME0plVDJccHAvcBKl83HgJu01onADcCHjR9xmTZBvnSMCGD9gezanZB0ExRkwdb  
/NWpcQojWqVUkyUXHD5FmD5N6ZCFEazAU2K21TtFaFwOzgCmVHPck8DxwahhWa71ea53meJoM+  
CulfBs7YFcDO4ax/uDx2h3ceSxEdIU1HzRuUEKIVqlVJMm2nMOk63B6S39kIYTnaw8cdHme6th2ilJqE  
NBBaz23mutcDqzTWhdVtlMpdatSao1Sak1GRsaZxnzKwA5hpOcWVb48dUUWi1mu+uBvkJ7cYDEIIQS  
0kiTZKz+dB1OXFiAu0MRQgi3UkpZgJeB+6s5pg9mlPm2qo7RWr+jtU7SWidFRUU1WHwDHN/41brk  
YsC1YPWFX14BrRssDiGEqFWSXNMkEKVUR6XUYqXUeqXUJqXUhY7tbp0EAkJAT4luRzV4cSEN  
um3hkII4Q6HgA4uz+Mc25yCgb7AEqXUPmA4MMdl8l4c8A1wvda6yfurJbQLwcfLURvJewABEabTxeYv  
4Ie/SqIshGgwXjUd4DIJ5FzM13arlVJztNZbXQ57FPhca/2WY4LIPCCeskgaUqpvACKnzt1+hOHAHgp  
G8Uvl7WJr21EEK4wWqgu1KqMyY5vhq4xrlTa50DRDqfK6WWAA9ordcopcKAucDDWuvlTRm0k4+X  
hT6xIaw/UMu6ZIAJj0JhDqz4J1i84ezHQKnGC1II0SrUZiS5NpNANBDieBwKpEHzmATiTJJ1UNsmva0  
QQriD1roUuAszKLENM4CRrJR6Qil1cQ2n3wV0Ax5TSm1w/EQ3csinGdghnM2Hciix2Wt3glJwwQsweA  
b88jKsfq9R4xNCtA61SZJrnAQCPA5MV0qlYkaR767kOIVOAmmsCSAAnDgMgFdIu4a9rhBCNFNa63la  
6x5a665a66cd2x7TWs+p5NhxFws1jsdPaa0DtdYDXH6ONnX8AzuGUvhiZ8eRE7U/yWKBSa+YjheLn4a  
C7EaLTwjROjTUxL1pwEytdRxwIfChY3IIUPMkkMaaAAKgHUMyb5umrfIQQghRP85FRepUcgEmUT7  
vSSg4DstfbfC4hBCtS22S5JomgQDcDHwOoLX+FfDDUfPm7kgxcfTKNLetGnT5N8YCiGEqIe4cH/ah/m  
zeEc9vlls1x8Sr4Tf3oKcin9VCSFE7dUmST41CUQp5YOZBFLxK7sDwNkASqnemCQ5ozlMAinMOkS6  
DqOdtH8TQogWQSnFpH7tWL0zg+z84rpFYMKjoO2w5JmGD04I0WrUmCTXchLI/cAtSqmNwKfADK2  
1phlMARHnHiadcNqF+TXlbYUQQpyBi/rFUmrxLEg+UveTwzvBkFtgwyeQsbPhgxNCtAq1qkmuaRKI1n  
qr1nqU1rq/Y6LHQsd2t08CsZw8QroOp32Yf1PeVgghxBno2z6ETm0C+Hbj4fpdYNS9ZjR56/8aNjAhRKv  
h8Svu+RVmkEE4kUGykIgQQrQUSiKm92vHij3HOJZX6crY1QuOgdiBsPuHhg9OCNEqeHaSXJSRHr+0k+  
b7RWC3SWF4IIVqSi/rHYtfw/eZ6jiZ3OwdSV5tuF0IIUueenSTnpQNgD4xxcyBCCCHqqmdMMN2ig/h2  
U32T5HNNycWexQ0bmBCiVfDsJNnRI9kSIqvtCSFES6OU4qJ+sazel8WRnMK6XyAuCfzCYPEPDR6bE  
MLzeXSSbM91LCQSIQuJCCFES3TxgFi0hi/WHKz54IosVug6wSTJWjd8cEIj+bRSfLJY6kABLWJc3Mk  
Qggh6qNzZCBndY/ko5X7KbHZ636B7uea0rsjmxs+OCGER/PoJLkgK5V87Utkm4Zd6loIIUTTmTEynvTc  
IuZvqUfP5K5nm9/S5UIIUcenSSX5hwmXYcRGy6r7QkhREs1vmc0ndoEMHPFvrqfHBwDbfvB7kUNH  
pcQwrN5dJKsThzhKOHEymp7QgjRYlksiutHxLN2/3E2pWbX/QLdz4UDv8GJeoxECyFaLY9Okn0KjnK  
MCEL9vd0dihBCiDMwNSmOQB9r/UaTB043k/h+/L8Gj0sI4bk8N0nWmsDiDPJ9o1BKFhIRQoiWLMTp  
mysGx/HdxsOk59axHVxEFxhxJ2z8BA6ubpwAhRAex3OT5JIC/HQhtoBId0cihBCiAdw8ugt2rXlj8e66n3z  
WAxDcDr7/E9jr0SVDCNHqeG6SXJAfGfdgGzcHloQQoiF0bBPA1KQOfLrqAKnH8+t2sm8QnPsEpK2H  
DR83ToBCCI/isUlyaV4mAD4hkiQLIYSnuOfsbiil+OeieowmJ06FuKGw9IWGD0wi4XE8Nkk+mXMMAJ  
8gKbcQQghP0S7Un2uHdeTLdansPXaybicrBb0nQ/YByM9qnACFEB7DY5Pk/JyjAPgEy0iyEEJ4kt+P64a  
P1cKrP+6s+8kxfc3v9C0NG5QQwuN4bJJclGvKLQLDZLU9IYTwJFHvswYFc+cjWlsO5xbt5OdSfIRSZ  
KFENXz2CS55IRJkoMkSRZCCI9z+5iuBPt68dKCHXU7MTgGAqMgPblxAhNCeAyPTZJt+Vnka19CQ4  
LdHYoQQogGFhrgzR3jurFo+1FW76tjfXF MX0jf3DiBCSE8hscmyeRnkU0gEYE+7o5ECCFEI5gxMp7oY

F+e/347WuvanxjTB45uB1tp4wUnhGjxPDZJthRlk0Mw/t5Wd4cihBCiEfj7WLn3nO6s2X+cxTuO1v7Etolg  
K4LMerSRE0K0Gh6bJHsXZXPSEixLUGhhhAe7MqkD8W0CeGnBztqPJsf0Mb+lw4UQohoemyT7luRQ4  
BXq7jCEEEI0Im+rhbsmgfr4dzajyZH9gSLtyTJQohqeWySHGDLpchbkmQhROujlJqolNqhlnqtlHq4mu  
MuV0pppVSSy7Y/O87boZQ6v2kiPjNTBsQSF+7Pa4t212402csHonpKGzghRLU8M0nWmkD7CUp9w9w  
diRBCNCmllBV4A7gASACmKaUSKjkuGLgXWOmyLQG4GugDTATedFyvWfO2WrhjXFc2HMxm+e7  
M2p0U00fafAkhquWZSXJxHt6UYvcLc3ckQgjR1IYCu7XWKVrrYmAWMKWS454EngcKXbZNAWZp  
rYu01nuB3Y7rNXtXDI6jbYgf//xpV+1OiOkLJ9JkeWohRJU8MknWjg89S0CEmyMRQogm1x446PI81bHt  
FKXUIKCD1npuXc9trny9rNw2tgsr92axam8tEl/n5L0j0i9ZCFE5j0yS87IzALAESpIshBCulFIW4GXg/jO8  
zq1KqTVKqTUZGRkNE9wZunpIRyICfZi5Ym/NB7dNNL+l5EIIUQWPPTJJPOpJkn6BIN0cihBBN7hDQw  
eV5nGObUzDQF1iilNoHDAfmOCbv1XTuKVrrd7TWSVrrpKioqAYMv/78faxcMTiOhcnpZJwoqv7goGgi  
jIaUxWC3N02AQogWxSOT5IlckyT7h0qSLIRodVYD3ZVSnZVSPpiJeHOcO7XWOVrrSK11vNY6HvgN  
uFhrvcZx3NVKKV+IVGegO7Cq6V9C/V01pAOlds2Xa1NrPnjoLbBrIXx7N9htjR+cEKJF8cgkufiEmd0cIE  
myEKKV0VqXAncBC4BtwOda62S11BNKqYtrODcZ+BzYCswH7tRat6jssWtUEMM6RzBr9QHs9hrawY  
35E4x9CNZ/BN/cJstUCyHK8XJ3AI2hNM8kycHh0W6ORAgmp7Weh4wr8K2x6o4dlyF508DTzdacE3g  
mmEduXfWBn5NyWRUt2oGS5SC8Y+Alx8s+j8IjYNzHm+yOIUQzZtHjiTb87PI036EhwS5OxQhhBBN7  
Pw+bQkL8OaTVQdq8JZf4T+18CK1yFzT+MGJ4RoMTwySbYUHiebIEL8PHKgXAghRDX8vK1cPiiOh  
clHOJZXwwQ+p3MeNyPK8//cqLEJIVoOz0ySi7LJU8EopdwihBCCDeYNrQjpXbNv5fXoh0cQHAMjHs  
Ydi2AnQsaNzghRIvgkUmyb3E2J60h7g5DCCGEm3SLDuLCxHbMXL6PrJPFTtp6K0Q2QPmPwwlhTUf  
L4TwAJ6ZJJfmUuQd6u4whBBCuNEfzu5OfomNd5el1O4ELx+44HnISoEPL4WTmY0boBCiWfPIJDnQlk  
uRT5i7wxBCCOFG3WOCmdwlv+s2EdmbWuTu06Ay9+HQ2vhvQmQ8jMsexneGg3/uUgWHhGiFfG8J  
NluJ1jnYfeVkJWQhhGjt7j27GwUlNt6p7WgyQOIVMGMuFJ+E/15s2sOVFsLepbD928YLtiEUn4SiE+6OQ  
giP4HFJsi7KwYod7R/h71CEEEK4WbfoYC7uH8t/V+wnt7Ck9id2GAK3LoGL/wn3boQ7V5p65cXP1rw6X  
/ZB2D4PdA2LmTSGL26E98+TFQSFaAAelyQX5poaMkuAJMlCCCHg2mGdKCixsWL3sbqdGBoHg66H  
8HiwWE33i4xtkPxN5ceXFMPL8DrQ2DWNPj+oaYtz8jPgt0/wtGtsPnLpruvEB7K45Lk3KyjAHgFtXFzJ  
EIIIzqDQR3DCPbz4uedGWd2oYRLIToBljxXfqRWa0ieDW8MhcVPQ4/zYMjvYNW/YM5dYCuBtA2w4  
p9wcPXp17XVYYs7OjvmgbZBYDT8/FzDLLO9/1dZrlu0Wh632kZBjhkp8A2WJFkIIQR4WS2M7hbJzzs  
y0FrXv4e+xQLj/gyfXwcLHoEOw8wCJMv/AQd/Mwn09f+DLuNM4hwYBUuehS1fQ2mBuYZvKNyyCC  
K7m+er34eFf4Vpn5jznDZ/CUW5kHRT7ePb9i2EdoQLnoNZ18DGT2HQdfV7rQAHVsK/J8KFL8HQW+p  
/HSFaKI8bSS7MNSMFAWFRbo5ECCFEczG2RxRpOYXsPpp3ZhfqfZFJZle+DV/eaMoqlLgon/AbcvKEI  
2ITHnGlDeg31S47F245SewesOn06AwBzZ9DnPvNwn0/+6GIkdsqWvg61th3oOQd7R2cRXmwp6fTHw9L  
4TYQab0o7SWPaIrs/lz83vL1/W/hhAtmMclycV5piY5UJJkIYQQDmN6mL8TzrjkQim4bjY8uBfu+BVu+A  
7uWQeDZ4C1ki9nB043k//6XQntB8NVH8Lxvaad3De3Q/xouO4byDkIi54wnSm+uhkC2oC9BNZ/WLu4di  
0EW7FJkpWC8X+BnAMw+3bYuRCK8+v2Om0lpvba6gMHfoXcw3U7XwgP4HFJsv1kFgCh4ZIkCyGE  
MGLD/OkRE3TmSTKYJDQgAmISoPNZ4Btc+3M7jYQLX4TDGyF2AEz71Iw+D73V1DB/fCVkH4Ar/w  
vxZ8GambXrVLFtDgTFmBIQgG5nQ9LNpsvGJ1Phhc6wcVbt40z5GfIzYfwjgDbXr46tBN4cAb+8Uvt7CN  
HMeVySrPOzyNUBhAb6uzsUIYQQzcjYHIGsTMkiv9jNE9GSboIbjUj0s4E++zHIKwjHFgBY/4EnUbAk  
JvNaPDuH6u/XnE+7PoBek02ddNgEvnJL8ND+8xIdfskmP172PZd7WLc/AX4hcHwO02tdVUdPZx2/WC6  
aix+1pSfOGXugR8eg9y02t1XiGbE45Jka+FxclQwXlaPe2lCCCHOwJgeURTb7KxMyXJ3KNB5DPiFID33  
DTKjx2fd2MeNNt6TTajw2s+qPo6uYdhxWtQkm9KLSry9jOrCF7zGbQfZOqoU5ZUH1txPmz/DhKmm  
KW6Ey6BA79VX3Kx8RPwjzA11/MfMdtOZsJHI5mJjW8Mh/UfnXnv6CObyyfhQjQij8skrcU5nLTU4asvI  
YQQrcKQ+Aj8vC0NU3LRGGIHmhFlZ22z1dv0ad65wJRgOGltul+8MQxe7mU6aMQkmvrmqvgGwTWf  
Q5tu8Ok1cHhT1cfunA/FeWblQYA+11Cu5CJICWz4tOz4/CzYMR/6T4OxD8LO702njc+mm8T68vehbV/4

353w8VQ4Wcd+1a73mTkJPrvOPQu1iFbH41rA+ZbkcmIaUvOBQgghWhU/bysjurRhyY6jaJ1Q/1ZwTWn  
wDFj2d5MY9rsS2g2ApS9CymJo2w/OfRK6jDVJsqWGca+ACFN68e4E0yLulsUQFGUWPFn1L9PLObit  
WX47uB10GmXOi+ppSi7WfWhKP3YtNNv9QqDXJJOW20tgwDVmVcL1H8HnN5iezZe/b5LtPpfB6ndN  
u7u3z4KpM6HjsLq9Fz8/b7qCFObAvl9MPbgQjcjjRpK9bQWUege6OwwhhBDN0IRe0ezLzGdPxkI3h1I7o  
XEw6e9QWmh6M8+80LSIu+BFs2z2qHugXf+aE2Sn4LZw9cdwMgM+vx5OpMOnV8H8h03i/esbkLbOJL  
wWa9l5CZdA+mbTO/ncJ02CPuceyMuADR9D20QzWuzIY/oqA4x9uGw02mKB YbfBzQvNCpNMC2HWt  
WU/B1ZWH/ex3bD6PTNa7R9hWvAJ0cg8byTZXkCpRSbtCSGEON3ZvWP46/+SWbQtnW7RQe4Op3aS  
bjI/x/fBwVWm60VIu/pfL3ag6d/81c3wj36me8aFL5IVAgEKs82kPVfDbwe/UJP0BkZC9/PgX2Pgkyvh8Aa  
Y+FzZsV3Gwp92m5Hr0+49AG5bCt8/aEauLVbIS4eDK+GOFRAUbY6zlZpWeRFdzDE/PGYWbjn3CQiJ  
hWUvm/cjPL7+74MQNfC4kWQfirBZ/dwdhhBCiGYoNsYfPrEh/Lgt3d2h1F14vCm5OJME2SnxCtNLOTw  
eblpgVtRTyvz4h5vfrvxCTaIcGGmeR/eCc/5mRp0tXpA4tfzxLSXITv5hcNk7cNcq+P2vMGOU6Q89+w5T+1  
GYYyb8vZ4EL3YzI8075sJZfzRJ9JDfgbLAqnfP/H0QohoelyT76SJXpIkCyGEqNw5vWNYu/84mXIf7g7  
FvcY+CHeuhLjB9Tt/2B1mdb8B15Qlz/UR3RvOf9rUOy/6P3j/fNi/HMY+BN3OMfXH4Z1h+O/N8SGxpvp  
Gug/LVikUohF4VrmF1vhRhM0rwN2RCCGEaKbO6R3DPxbtYvGODK4YHOfcFoui8UshtIQkm6G3Ytg  
+avgGwrTvypb4ttWCtpu6p2dht8ByV/Dwr+YUhGrt9lefNJ01Ijs1jBxiVbNs0aSS4uwoLFLuYUQQogq9G0  
fQkyILz9ubYEIf55KKVMnPfRWuHIBWYIMpiWea4IMEDfELHSydqZZ4jv7oJnY99pAeGOoqVcW4gx5  
VpJcYtam114ycU8IIUTllFKc0zuGpbpsyKCypxZLPomkERJglu6N713ysUjDxGdNi7vBGeDUR5t5vVi3UN  
rNioBBnqFZJslJqolJqh1Jqt1Lq4Ur2d1RKLVZKrVdKbVJKXeiy78+O83Yopc5vyOAr0sWmpY/2lnILYQ  
QVTsnIYb8Yhu/pmS6OxRxJhKvgFt+MhMHp30GN/8AHUfCps9lwRFxxmpMkpVSVuAN4AIgAZimlEq  
ocNijwOda64HA1cCbjnMTHM/7ABOBNx3XaxSIRY6RZG8ZSRZCCFG1EV3aEOBjlZILTxDdGy5/F3p  
ONCPM/abCsZ1mhLm2Dq01fZ/t8s2CKFObkeShwG6tdYrWuhiYBUypcIwGnMvchQJpjsdTgFla6yKt9V5  
gt+N6jaK00DHLVUaShRBCVMPP28rYHIes3JqO3S4jh4l4RKweNet5GLZy7DuP3B0a6OFJVqe2iTJ7Y  
GDLs9THdtcPQ5MV0qlAvOAu+twboMpKXSsoCRJshBCiBpM7NuWjBNFrDtw3N2hiIYUEAHdz3UsI12  
LkeGCbNj1g3m8/9dGDU20LA01cW8aMFNrHQdcCHyoIKr1tZVStyql1ii1mRkZNQ7CJu3EL5yLLUQoj  
WqxbzSG5XSm1WSm1QSv3iLKFTSnkrpf7j2LdNKfXnpo++6UzoFY2P1cL8LUfcHYpoaP2uhLwjsG9Zz  
cdw/w5sRWD1hQOSJIsytUlkDwEdXJ7HOba5uhn4HEBr/SvgB0TW8ly01u9orZO01klRUVG1j74CZ7mF8  
pGaZCFE61TLeSSfaK0TtdYDgBeAlx3bpwK+WutEYDBwm1IqvkkCd4NgP29Gd4/k+y1H0DLJy7P0mA  
g+wWYCX002f2kWK+l9kUmS5b8F4VCbJHk10F0p1Vkp5YOZiDenwjEHgLMBIFK9MULyhuO4q5VSvk  
qpzkB3YFVDBV+RzdHdwuIrI8ICiFarxnkkWutcl6eBmHklOH4HKqW8AH+gGHA91uNM7NOWQ9kFb  
Dnk0S+z9fH2NxP4NnxieilXJe8o7P0Z+l4OnUbAicPSY1mcUmOSrLUuBe4CFgDbMF0skpVSTyilLnYcdj  
9wi1JqI/ApMEMbyZgR5q3AfOBOrXWjTR21FzmSZCm3EEK0XrWaC6KUulMptQczknyPY/OXwEngM  
Gbw4yWtdVZlN2moMj13OzchBqtFMT/5sLtDEQ3t/GfMstbf3gu/vVX5McnfmNX8EqdCxxFmW1UlF0V5  
MsrytSqbhrPU9r3UNr3VVr/bRj22Na6zmOx1u11qO01v211gO01gtdzn3acV5PrfX3jfMyDHuxqUn28pOJ  
e0IIUR2t9Rta667AQ5g2nmBGoW1ALNAZuF8p1aWK8xukTM7dwgN9GN4lQkouPJG3P1z9MfSaDPMfh  
vmPQNGJ8sds/hJi+kJ0L4jqDX5hlSfJqWvgxa6w/qMmCV00Dx614p52JskykiyEaL1qNRfExSzgEsfja4D5  
WusSrfVRYDmQ1BhBNicT+7YjJeMku4/muTsU0dC8fGHqfyDpJvjtDfhnkkI0k2ebpD11lSm1ALBYoOPw  
0ztc5B2Fz66D0kLYMa/JX4JwH49Lkku1BW8fX3eHloQQ7ILjPBKIVHeXp5OAXY7HB4AJjmMCgeHA9  
kaP2M3OT4hBKZizMa3mg0XLY/WCya/A7xZBaHv4353wxQ2w+j3oMg4GXV92bMfhkLkL8hwIRLYS+  
GIGFBYH+LNg7zKwlbrjVQg38HJ3AA2qpIB8fPHxbrRF/YQQolnTWpcqpZzzSKzAB855JMAaR5ncXUq  
pc4AS4Dhwg+P0N4B/K6WSAQX8W2u9qelfRdOKDvFjfM9oZq0+yN0TuuPj5VHjR8IpLglu/hH2LgHfU  
GibCF4+5Y/pONL8PvibqWf+/kHYvxwufceMSn9xg1mdr+OwJg9fND0PS5LzKcRXPuCEEK2a1noeZmE  
n122PuTy+t4rz8jBt4Fqd64Z34saZq1mQfISL+se6OxzRWCwW6Dqh6v2xA8DLD1b+y5Rj5ByAUX+A/l

BfhagIGWxJMmthEdlk6oknwLtg4/Vo16WEEKIRja2RxQdIwL48Lf97g5FuJOXL7QfbBYh8Q2GGXPh3P  
8z+wIiTBK9Z7FbQxRNx6OySeUst5CRZCGEEHVgsSimD+/Iqr1ZbD8iPZNbtfofgUv/BbcthfjR5fd1GQ+  
pq6FQ/htpDTwqm1S2Aim3EEIIUS9TB3fA18vCh7/KaHKrFjsA+l9tJvxV1HU8aJupUxYez6OySWtpAQX  
aB18vmbgnhBCibsIDfbiofyzfrD9EbmGJu8MRzVGHYeDILyUXrYTHJcn5+OIrI8ICCCHqYcbIePKLbby3  
NMXdoYjmyMsXOo2ElCXujkQ0AY/KJq3OcguzCeEEKle+rYP5eL+sfraQqpx/PdHY5ojrqOh2M7IPtg  
zceKFs2jskkvWyGF+GKxKHeHl0QQooV6+IJeKAXPfe/x66iI+ug1GSxe8MvLDXO9b26Hz6Y3zLVEg/K  
sJNleSJHFz91hCCGEaMFiw/y5bUxXvtt0mNX7stwdjmhuIjpD0s2wdiYc3XZm10rfChs/hW3fwYn0BglPN  
ByPSpK9bQWUKFmSWgghxJm5fWxX2oX68X/fJINcand3OKK5Gfew6aO88K9ndp1fXgarD6Bh+7cNEp  
poOJ6TJNvteOtiSmQkWQghxBny97Hy18kJbDmUyx8/34DNrt0dkmhOAiJgzJ9g9w+w6wfYOgdmToYPL  
3OszFcLWSmw5SsYeitE9oCt/2uY2LL2gpb/XhuC5yTJpQUAFEuSLIQQogFcmNiORy7sxXebDvPo7C1o  
STyEq6G3Qng8fHwFfH4dHN9nVur7YCJkH6j5/F9eBYs3jLwbEqbAvl8gL6P+8djtsOAv8NoASP66/tcRp3  
hOklxsZiHbrJlkCyGEaBi3junKXeO78emqA7y4YIe7wxHNiZcvXPSamch31cdw70aY/jWcOALvnQs/vwi  
LnoAf/mZGjV3lpsGGT2DgdAhua5JkbYft39UvlpIC+HIG/Po6oKRFXQOpZDmZFqrEJMkl1gA3ByKEE  
MKT3H9eDzJPFvPmkj2M7BrJ6O6R7g5JNBddxpofp85nwU3z4dOrYPFToCyAMgnxdd9A274mif78epM  
Uj7rHnBfTFyK6mpKLpBvrFoPdBh9PNNSPR5z0Ne5fC/18b7CW2Zp4zklxiyi1kJFkIIURDUkrxt4sS6BIVyI  
NfbpTV+ET1YhLgng3w6FF4LAt+/5tpGTfzQljzAfxrrOlqMfXfplwDQCkzmr3KZzMrNv9Nnxsyjwu+ge  
MvAs6jYDMXXDyWNkxKT/D4mca6hW2Gh6UJJ8EwG71d3MgQgghPI2ft5W/T+3PkdxCnvpuq7vDEc2  
dxWrKMZSCqB5mdDmgDXx3H3j7we9+NEmxq4QpoG3w6z+h6ETt7lOYa0o6OgyDQdebbR1HmN8Hfis  
7bslz8PPzkHv4zF9bK+JBSbJjJNlkmQhhBANb2DHcG4f25XP16SyaJv0tBV1EN4JbpwP5zwOtyw2o80V  
tesPHUfCL6/Ai93gs+tq7p287CU4mQETnzUJOUDsQLD6wgFHyUXu4bLHe39usJfUGnhOkuyYuKe9JUk  
WQgjROO49pzu924Vw76wNbErNdnc4oiUJjoHR95n2cZVRCmbMNcn0oBtg53wz+luVrBT47S3ofw20H  
1y23csX2g8qG0neNgfQ4OUPexY32MtpDTwnSXZM3JNyCyGEEI3F18vKv2cMISzAmxs+WMWu9Fp+  
LS5EbVgspqb4whcg8UqzGl9Bdt+9R+bUebnOsGbI0wLubMfO/06HUfA4Q1mADH5G4hOgJ4XmK4X0  
sqw1jwoSTblFtpbulsIiYRoPG1D/fjo5mF4WS1Mf38IKR157g5JeKJht5oBwPUfmed5R+H7hyAkFvpdBYN  
vhGmfQEi708/tOALspbB9rhlR7nMpdB0PeUfOfCntVsSDkmQzcU9JuYUQQohGFh8ZyEc3D6O41M6UN  
5bzw1apURYNzFmjvOod0+Zt0RNQWghX/NuMNF/wHHQZV/m5HYYAChb9H6Ah4ZKyY6WHcq15UJ  
JsRpLxkZFkIYQQja9n22Dm3DWa+DaB3PLfNby4YLssXy0a1rDbIHu/qU1e/xEMvx3adK35PP9wU2Krc  
9D8juoBYR1NL+YUqUuuLQ9MkmUkWQghRNPoEBHAF7eP4OohHXhj8R6mv7eSoycK3R2W8BS9jk  
NIe5MkB7SBMX+q/bkdh5vffS4t29Z1POxbDqXFDRunh/KcJLn4JMXaC29vH3dHl0QQohXx87by3OX9e  
PGKfqw/eJwL//ELK/Ycq/IEIWpi9YIhvzOPz34M/EJrf263c8DqA30vL9vWZbwpT01dVf7YohOQuUcm9V  
XgMctS65J8CvDB1+o5eb8QQoiWY2pSB/rFhXHHx2u59r2VXD2kIw+e35PwQBm8EWdg+O+hTTczqlw  
XPS+AB3aBf1jZts5nmaWyl/8DtnwNGTvM6nx5jpr6SS/DkJsbLPSWzmMySntxAfn44ePlMS9JCCFEC9O  
zbTDf3jWa343uzOdrDjLh70uYszHN3WGJlszbDxIuNu3h6kKp8gkymJHoTqNg10LY/CXYS6DbuXD238  
yqfYuegPysBgu9nOJ8eO9cSJ7dONdvBB4zkmwvOkmB9pEkWQghhFsF+nr xl0kJXD44jke+3sy9s9ZjUTC  
5X6y7QxMCps0y5RXBbctW6QMz8vzWKPjpSZj8SsPfd80HpsxjeSn0uaThr98IPCaj1CX5FOKLj5RbCCG  
EaAZ6tQ3hk1uGM6RTBH/8bCMrdkudsmgGfINMb2XXBBkgujcMvRXW/BsOb2zYexbnw/JXwTsA0tZB  
enLDXr+ReExGqYtOko8vvt5Wd4cihBBCAGZS37vXJ9E5MpBbP1zLzOV7+Xz1QeZsTCOvqNTd4QIR3  
riHTReN7/4Ih9aBrYH+G13zPpzMgMvfM6sErv+4Ya7rdHhjoyTeHpMkU1Jgyi1kJFkI0coppSYqpXYopX  
YrpR6uZP/tSqnNSqkNSqlflFIJLvv6KaV+VUolO47xa9roPU9ogDf/uWkoEYE+PP7tVh78ahP3fLqeK95aw  
eGcAneHJ0QZ/zCY+CwcWgvvjofnOsCn0yBtQ/2vWXzSTBTsMh56TYJeF8KmWZW3obOVQGFu3a6v  
Ncx9AD6+suGSegfPyShL8inAV2qShRCTmlLKCrwBXAAkANNck2CHT7TWiVrrAcALwMuOc72Aj4Db  
tdZ9gHFASROF7tHahvrx4x/HsvKRs/nlofG8c91gUo8XcNmbK9iZfsLd4QIRpt+V8MetZmW/gdPhwK/wzl

j4YgYc31/36612jCKP+7N5PvA6yM+EnfPLH5e5B94aCW+PhtKi2l9/1w+m1nnM/aZIXgPynIxSkmQhhA  
AYCuzWWqdorYuBWcAU1wO01q5DNYGAsznqecAmrfVGx3GZWmtbE8TcKvh4WYgJ8SMuPIDz+rT1  
s9uGY7NrLntzBXd+so4PftnLjiOSMItmICQW+l4GF74I9240i5jsXAgfX2FGe2vDbodlL8OPj5uezR2Hme1  
dJ0BwrFlB0GnPnh3AuQcMisMbvy0bJ/WsOP7yrtu2O1momF4vEm+G5jHdLdQpabcIkSSZCFE69YeOO  
jyPBUYVvEgpdSdwB8BH2CCY3MPQCulFgBRwCyt9QuV3UQpdStwK0DHjh0bLPjWpE9sKF//fiQvLdj  
B6n3HmbvpMACT+7Xj4Qt6ERce4OYIhcC0jZvwKMQOhFnXwNqZMPQWs+/gKvhsOmg7BMWYjhmR  
Pc0y2Fv/B3t+Miv+XfSPsutZrDBgGvzyCnx0OeQdNfXEkT1g2qfw5Y3wy6swYLoZGd7yFXx1M/ScBNM  
+KR/b9m/hyCa45G2wejf4S/eYJNISWkABvrKYiBBC1ILW+g3gDaXUNcCjwA2YvxNGA0OAfGCRUmq  
t1npRJee/A7wDkJJSUJM01VNceACvXj0QgMM5BXy66iDvLN3Dwq3pTB/WiSsGx5EQG+LmKIUAel4I  
nUbDkmdNSYat1JRgWH2h29km2c056Fj2ugC8/GDyzB4xumdNAbPgJ0LzOhwcDvoPAbGPgR+IXDW/  
SbxTv4G4kfD3PvBJxh2zIWUJdBlnLmG3QaLnzHJdb8rG+Ule1ySLOUWQohW7hDQweV5nGNbVWYB  
bzkepwJLldbHAJRS84BBwGlJsmh47UL9+eO5Pbh6SAdeXLCD//66jw+W76VX22C6xwSj tcbLorh8cBxn  
dY9yd7iitVEKzn8K3hkHS1+Co1tNrfHNC80os5PdDjkHwMsfgmMqv1ZYR7hjeeX7ek6CqN6w7O+w+X  
NTn3zLIvj0apj/CNy21IxGL3sZMrbD1JnmeSPwjCTZVoLFXiKLiQghBKwGuiulOmOS46uBa1wPUEp11  
1rvcjydBDgfLwAeVEoFAMXAWKARVhUQ1YkN8+eVqwbw18kJzN2UxpyNaWxOzUYpRW5BCbM3p  
DEpsR1/mdQbq0WReryA3MISQvy8CPHzpn24PwE+nvHXu2hmYgdCv6thxWvm+aS/l0+QwawMGB5f/3  
tYLHDWH+HrWyBjG0x8HmL6wLIPwhc3wOr34Mhm2PARJFwCvafUeMn68oz/i0pMC518fPH1kj7JQoj  
WS2tdqpS6C5PwWoEPtNbJSqkngDVa6znAXUqpczCdK45jSi3QWh9XSr2MSbQ1ME9rPdctL0QQEejDd  
SPiuW5E/KlthSU23l2awuuLdzN38+FKzwv29eLqoR2YMa0z7cP8myha0Wqc/RjsmAc9JkLSzY1zjz6XmZ  
rl4LZmgROAhCnQcSTMf8g8H/sQjH247st114GHJMn5AGbFPRIJFkK0clrrecC8Ctsec3l8bzXnfoRpAyeal  
T9vK3ef3Z1LBrbnm/WHCA8wI8eh/j7kFZWSnV/Mom1H+WD5Pj5Yvo/4NgGEBfgQHuDD1KQ4zkuIQ  
VWsDxWiLkLbwx82gV/Y6bXGDcXqBbf8ZOqdnUmwUnDhC/D1rabbRt/LGufeLjwqSZZyCyGEEK1Bh  
4gA7jm7e6X7pgxoz8MX9OKTIQfYe+wk2QXFbDucy20frmVofAR3TuhGTkEJ2w7nkplXRHSwHzEhvo  
QF+BDgY8Xfx0r/uDACfT0jRRCNwD+88e/hXcm3IG0T4fe/Nv69HTzj/4BikyTn4ycr7gkhGj1YsP8eeD8  
nqeel9rsfLbmIK/8sJMbPlgFgJdFER7oQ9bJYmz28g1K2oX68cSUvpybUDbxym7XWCwyCi1aD89Ikh01y  
YXISLIQQghRkZfVwrXDOjFIQHtW7D5Gh4gAukYF4eNlwWbXZOYVkv1QQkGxjaMnivj7wh3c8t81n  
N0rGm+rhU2p2WQXIPDnC3szfVhHKdkQrYKHJMn/396dx1Vd5Y8ffx32VWRzBQV3JbyyuC9gWFkyk  
NsUZblMVrY4Wk1De9PUfJvq16Qz5mRqRWPYaOJomeSu5YpL5UYuYOJCiIoiKss9vz/u9QYIagpe78f3  
8/HgEZ/zWe773HM5vT33fD7nwnQLd9wlSRZCCCFq5OPuwu0RTaqUOTspGjXwoFEDD1tZfPtgPlyzny  
nL9xLk605MWADHz5znpfnbWb+vkCcT2rBidwFf/XiY8gpN37ZB9G0bTGRzP/y93a53tYSof4ZKkktwl+  
kWQgghxDVydXbisfg2jItrbRs1Nps109bs5+3MbNuTNaJaNKSBlhzOfuD3Ah2tyAPDzdCU8yJu4dsEkdm5  
K28a+dquHENfCUElymXKX+VJCCCFEHak8rcLJSfFoXGt6tgpk688nSOjYmNAAy9LZZ0sr2JR7nJ/yT5  
Nz7Ay7j55m8vI9TFq2h3aNfUju0pwkUzPb8UI4AoMkyZY5yeUu8jxIIYQQoj6ZQhtiCm1YpczTzZl+7YL  
p1+7XIQB/OXWOxTuOsvD7w7ydmc3bmdl0aOJLy0AvWgR40adtMP3aBsn8ZnHDMkaSbH26hdnZ4zIH  
CiGEEOJ6aNTAgwd7hvFgzzDyTpSw4PvDbMo5zr6CM6zMLuDDNTn0aRPEc3d1oFPTBpw6V865sgoa+  
bpL4ixuCMZIkq3TLSpkJFkIIYS44YT4e/FYfBuIt2yXlpuZteEAK5btYdDkb3F2UrbH0EU0a8ADPSxP4vB  
0k1V0hf0YKknWLjKSLIQQQtzo3FycGN07nCHRIfxn/QFKSsvx93JDa/hiSx6p837ktS93EtWiITEt/OkWH  
khsmD8erpI0i+vHMElyqXLH1cUY1RFCCCFuBn6erjzev02Vs0f6hpN14AQLvz/M5gMn+NeKvZiX78XD  
1YkerQJpFeSDUpbFUG6PaExMywA7RS+MzhhZZdlZzit33FzkX5hCCCGEI1NK0TUsgK5hluS3+Hw5m  
3KOs+qnAlbvKSAr9wRaa0orzHywej933tKEPw/sQFfQt50jF0ZjjCQ5PI5vfiqT1faEEEIIg/Fxd6F/h0b079C  
oSnlJaTkfrs7hg9X7WLIzn7simzKyV0uiW/jLjX+iThgjSe6UxJxvg3E32zsQIYQQQlwPXm4u/HFAW1K6h  
/LBqv38N+sgC74/zC3NG/BYfBsGRjSRtRPENTHM0GtpuRI3V8NURwghhBBXoJGvBy8ldmLD8wn8bX

AkJecreGzWFm5/bzWzN/7MyZJSe4coHJRhssrSCrMsSS2EEELcpLzcXLivewuWPBXHv+6LwsVJkTrvR  
2JeX8oDMzYwf+shyiosXzmbzZr/bTvEmI838VP+aTtHLm5UxphugWUkWeYkCyGEEDc3ZydFYudmDI  
psyvZDp1i0/QiLfjzChM+38XZmNvd1b0HmjP8kFeEk4Kdh08xd1xPQvxlyWxR1RVIIUqpgUqpbKXUXq  
VUag37/6GU2mb9+UkpdLSvreUUjuUUrUUUpNVpc2mlyRZCCGEEBcopYgM8ePPAzuw4ul4pj8YS1M  
/D97OzObY6fP8v+EmFj7ZhzoI5Tw4YyOFxec5UHiGGd/m8MXmPHuHL24Alx1JVko5A1OA24A8YJN  
SaoHWeueFY7TWEysd/yQQZf29F9Ab6Gzd/S0QB6ys0/htSstluoUQQgghLubkpBjQqTEDOjUm99gZmvh  
52BYmmTmqKyOmb6DfWys4U1phO8fFWZHcpbm9QhY3gCvJKrsBe7XW+7XWpcBsIPkSx6cA6dbfNe  
ABuAHugCuQf/Xh1q60QkaShRBCCCHFpYUHeVVbu6xoWwAcPxNCzdSCv/K4Ty5+Oo1t4AM/O/YhvD  
560X6DC7q5kTnJz4GCI7Tyge00HKqVaAuHAcgCt9Tql1ArgCKCAF2mtd9Vw3sPAwwAtWrT4LfHbnC+  
TJFkIIYQQv118+0bEt//1OcxT748mecp3jE3L4o3BkZw6W8apc2XceUtTmvh52DFScT3V9Y179wJztdYV  
AEqpNkBHIMS6f4lSqq/Wek3lk7TW04BpALGxsfpqXvi8jCSLeJWVkZeXh7nzp2zdyjiBuLh4UFISAiurq  
72DkUIUccCfdyZPjKWoe+vZWxalq3836v2MWNkV25p7mfH6MT1ciVJ8iEgtNJ2iLWsJvcCj1faHgys11o  
XAyilvgZ6AmtqOPeqaa0tz0mWZalFPcjLy8PX15ewsDBZxUkAlj6nsLCQvLw8wsPD7R2OEKIedGjSgC  
VPxXHo5FmCfdw5da6Mcf/ZwvB/r+OfKVEM6NTY3iGKenYIQ6+bgLZKqXClIBuWRHhB9YOUUh0Af  
2BdpeKfgTillItSyhXLTxsXTbe4VmUVlsFndxlJFvXg3LlzBAYGSoIsbJRSBAYGyrcLQhhcs4aedA0LICzI  
m84hDcl4vBdtG/sw9tMsHvoki/X7C9FaU1Zh5ufCEopKyuwdsqhDlx1J1lqXK6WeADIBZ2Cm1nqHUuo1I  
EtrfSFhvheYrbWuPF1iLnAr8COWm/gWa60X1mkNsNy0B8jTLUS9kQRZVCefCSFuPo18Pfj84Z5MXbW  
P/6w/wL3T8gnyceNESRkVZo2vuuvJEUwNLq59BEGcEVzkrXWi4BF1cperrb9ag3nVQCPXEN8V6S03  
Joky0iyMKDCwkISEhIAOHr0KM7OzgQHBwOwceNG3Nzcaj03KyuLtLQ0Jk+efMnX6NWrF2vXrq2zm  
CdMmMCcOXM4ePAgTk7ydymEMA5PN2eeuq0dj8W3Zt6WQ2QdOE7zhp40b+jJvC2HeGbO93yz4yipd  
3YgPMhbkmUHZogV9yRJFkYWGBjItm3bAHj11Vfx8fHhmWeese0vLy/HxaXmP+XY2FhiY2Mv+xp1m  
SCbzWYyMjIIDQ111apV9O/fv86uXdml6i2EEPXNw9WZ+7q34L7uvz6Va3hsKDO/zeHtzGy+2ZlPkwyEd  
G8VwICOjUno2AgvN+mzHIkhskpbkizTlcRNYtSoUTz66KN0796dZ599lo0bN9KzZ0+ioqLo1asX2dnZA  
KxcuZLExETAkmCPGTOG+Ph4WrVqVWW02cfHx3Z8fHw8w4YNo0OHDtx//1cmEG1aNEiOnToQEx  
MDOPHj7ddt7qVK1cSERHBuHHjSE9Pt5Xn5+czePBgTCYTJpPJlpinpaXRuXNnTCYTDzzwgK1+c+fOr  
TG+vn37kpSURKdOnQC4++67iYmJISIlgmnTptnOWbx4MdHR0ZhMJhISEjCbzbRt25aCggLAksy3adPG  
ti2EENfK2Ukxtl8rlj8Tx+t330JsmD/f7T3Gk+lbifnrUsanb+VA4Zkq5xSVILF+fyFf/3iE9I0/X7Rf2I8h/klzvty  
yQo6MJIv69peFO9h5+FSdXrNTswa88ruI33xeXl4ea9euxdnZmVOnTrFmzRpcXFxYunQpzz//PF988cVF5  
+zevZsVK1Zw+vRp2rdvz7hx4y56hNnWrVvZsWMHzZo1o3fv3nz33XfExsbyyCOPsHr1asLDw0lJSak1rv  
T0dFJSUkhOTub555+nrKwMV1dXxo8fT1xcHBkZGVRUVFBcXMyOHTt4/fXXWbt2LUFBQRw/fvyy9d  
6yZQvbt2+3PVVi5syZBAQEcpbsWbp27crQoUMxm82MHTvWFu/x48dxenJixIgRzJo1iwkTJrB06VJMjP  
Nt6oqRKKUGApOw3EcyXWv9ZrX9j2J5EIEFU Aw8XHkVVaVUC2An8KrW+p3rFrgQBhHi78WIHi0Z0  
aMIFWbNxpzjfPnDYf637Tdf7DzKM7e3J7ILc2Z8m0PaulxKKq305+XmzFvDOpPYuZkdayDAMEmyTL  
cQN5/hw4fj7Gx57GFRUREjR45kz549KKUoK6v5DutBgbwh7u6Ou7s7jRo1Ij8/n5CQkCrHdOvWzVbWp  
UsXcnNz8fHxoVWrVrbENCULpcqo7QWlpaUsWrSID999F19fX7p3705mZiaJiYksX76ctLQ0AJydnfHz8y  
MtLY3hw4cTFBQEQA BwGXr3a1btyqPXZs8eTIZGRkAHDx4kD179IBQUEC/fv1sx1247pgxY0hOTmb  
ChAnMnDmT0aNhx/b1HI1SyhmYAtyGZfGnTUqpBZWTYOAzrfW/rccnAe8CAyvtfx4+jqFLIShOTsper  
YOpGfrQJ64tQ0vZGzn9a928cYiy8O+fte5GcNiQgjyccfJCv7I2M4Tn21184ETPH17e3zcDZGqOSRDvPM  
Xnm4hj4AT9e1qRnzri7e3t+331156if79+5ORkUFubi7x8fE1nuPu7m773dnZmfLy8qs6pjaZmZmcPHmSyM  
hIAEpKSvD09Kx1akZtXFxcMJstf9dms5nS0lLbvsr1XrlyJUuXLmXduV4eXkRHx9/yceyhYaG0rhxY5Yv  
X87GjRuZNWvWb4rLQXQD9mqt9wMopWYDyVhGhgHQWlf+OsQby9OHsB5/N5ADyHe+QtSxp6ez  
BgZy8IfjrDt55Pc1z2UNo18qxwz++Ee/N+i3cz8LofPNvzMrR0akdylGbd3aoKTk+UmwLIKM/9ctoejp87x+

9hQYlr6yw2C9cAQWaXcuCdudkVFRTRv3hyAjz/+uM6v3759e/bv309ubi4An3/+eY3HpaenM336dHJzc8  
nNzSUnJ4cIS5ZQUIJCQkICU6dOBaCiooKioiJuvfVW5syZQ2FhIYBtukVYWBibN28GYMGCBbWOjBc  
VFeHv74+Xlxe7d+9m/fr1APTo0YPVq1eTk5NT5boADz30ECNGjKgyEm8wzYGDlbbzrGVVKKUeV0rt  
A94CxlvLfIA/A3+5DnEKcVNSSpFkasbLv+t0UYIM4OrsxMu/68S8x3pxb9dQNuWe4NH/bOHeD9ezv6C  
YwuLzjJi+gcnL9/LID0cY9u913PHeajK25mE2X9WixaIWhsgqlTJMpIslblPPvsszz33HFFRUb9p5PdKe  
Xp68v777zNw4EBiYmLw9fXFz6/qsqwIJSUsXryYQYMG2cq8vb3p06cPCxcuZNKkSaxYsYLIyEhiYmL  
YuXMnERERvPDCC8TFxWEymXjqqqacAGDt2LKtWrcJkMrFu3boqo8eVDRw4kPLycjp27Ehqaio9evQAI  
Dg4mGnTpjFkyBBMJhP33HOP7ZykpCSKi4sNOdXit9BaT9Fat8aSFL9oLX4V+MeFVVIVRSn1sFIqSym  
VJTc/ClH3olv485fkW9jwfAJvDe3M7iOnGDhpDXdOWsO2gyd5754uZL04gL8PjcTZyYmJn3/PkKlr2frzC  
XuHbhiq6tof9hcbG6uzsrIuf2AlS3fm81BaFguf6ENkiKynLurWrI276Nixo73DsLvi4mJ8fHzQWvP444/Ttm  
1bJk6caO+wfrOsrCwmTpzImjVrrvlaNX02lFKbtdaXf+5ePVFK9cRyw90d1u3nALTW/1fL8U7ACa21n1Jq  
DRBq3dUQMAMva63/danXvJp+Wwjx2/xy+hx/WbCTHYeLmJwSReeQhrZ9ZrPmiy15vJWZTcHp88S29  
Ce5SzPuimxKoI977RcVl+yzDTUnWaZbCFF/PvzwQz755BNKS0uJiorikUfqfZ2gOvfmm28ydepUo85Fvm  
AT0FYpFQ4cwrIa6n2VD1BKtdVa77FuDgL2AGit+1Y65IWg+HIJshDi+mjk68GU+6Nr3OfkpBgeG8qdkU  
35dN0BMrbm8dL/dvD6V7v44IEY4ts3us7RGomXkmSZkyxEvZs4caJDjhXlpqaSmpqqr3DqFda63KI1BN  
AJpZHwM3UWu9QSr0GZGmtFwBPKKUGAGXACWck/SIWQtQVH3cXxsW3Zlx8a3YdOcXT//2eRz7d  
zMeju9GzdSD5p87x7jc/kXeyhNbBPrRp5MPtnZrQxM/D3qHfkAyRJMtzkoUQ4lda60XAomplL1f6/Y9XcI  
1X6z4yIcT10rFpAz79QzfunbaeP3yyiRE9WjJr/QHKKjTtm/gyb8shis+X87dFu3g0rjWP9GuNp5shb2a+aoZ  
IkmXFPSGEKEIqgJ93Jn1UHd+/8E6pq3ez4COjXgpsRMtA73RWp/2Bne/eYn3lu6h883HWRM73CGxY  
Tg7+2G1pqcY2dwUoqwoJpvnjY6QyTJFxYTcXeVJFkIYQQ4oJGDTz4YlwvcgtLiGnpbytXStE62Icp90cz  
Muc4by3ezRuLdvH2N9l0DfMn++hpjhWXohT8oXc4T9/e/qYbaTZEkmy7cU9GkoUQQgghqgj0cb/kUy66  
hQ cwd1wvdh05xWcbfmZjznH6tQ2ma3gA2w8VMf3bHJbsye5S3NOnCml8Mx5is9XcK6sgrIKM4183Qn1  
96JZQ08aernSwMOV8GBvWgf7XMda1j1jJMky3UIYWP/+/UINTeWOO+6wlb333ntkZ2fbFueoLj4+nnfe  
eYfY2FjuuuusuPvvsMxo2bFjlmFdffRUfHx+eeeeaZWl97/vz5tGvXjk6dOgHw8ssv069fPwYMGHDtFQMmT  
JjAnDlzOHjwIE5O8vcrhBD21LFpA/569y0XIQ/q3JTn5v3I5GV7aOjlSqC3G74erni4OuHt5sK+gjOszC6wf  
bN/QbfwAB7s2ZIBHRvj4ep4o9CGSZJdnZVtuUYhjCQIYXZs2dXSZJnz57NW2+9dUXnL1q06PIH1WL  
+/PkkJibakuTXXnvtqq9VndlsJiMjg9DQUFatWkX//v3r7NqVIZeX4+JiiK5OCCHsolfrIFY8HU+5Wdf6kAS  
tNSdKyjh1toyis2Ws31/IfzYc4InPtgLg7+VK4wYeuLk4UV5hWaMj2NedZg09aBHgTa/WgdzS3A9nJ8X58  
gr25BdzvtxMAw8X/DxdCfZ1v+5Lbxti6Ka03CyjyMKwhg0bxldffUVpaSkAubm5HD58mL59+zJu3DhiY2  
OJiIjglVdeqfH8sLAwjh07BsAbb7xBu3bt6NOnD9nZ2bZjPvzwQ7p27YrJZGLo0KGULJSwdu1aFixYwJ/+9  
Ce6dOnCvn37GDVqFHPnzgVg2bJlREVFERkZyZgxYzh//rzt9V555RWio6OJjIxk9+7dNca1cuVKIiIiGDD  
uHOnp6bby/Px8Bg8ejMlkwmQysXbtWgDS0tLo3LkzJpOJBx54AKBKPAAPj62a/ft25ekpCRbgn/33XcT  
ExNDREQE06ZNs52zePFioqOjMZlMJCQkYDabdu2LRdWkTObzRp0wZZVU4IcTNzclKXfIqYUooAb  
zfCgrwxhTbkkbjWrHymPx+N7sozt7djUOemhPh7EujtRrOGnjT18+D4mVKW7Mzn74t3kzzlO2JeX8Lt/lh  
FxMuZJP7zW4ZOXctt/1hNt78to9/bK/jbol1sPnCcM+frfmXZmhieKW0wiypfxPXx9epcPTHur1mk0i4881  
adwcEBNCtWze+/vprkpOTmT17Nr//e9RSvHGG28QEBBARUUFCQkJ/PDDD3Tu3LnG62zevJnZs2ezbd  
s2ysvLiY6OJiYmBoAhQ4YwdxYAF588UVmzJjBk08+SVJSEomJiQwbNqzKtc6dO8eoUaNytmwZ7dq1  
48EHH2Tq1KIMmDABgKCgILZs2cL777/PO++8w/Tp0y+KJz09nZSUfJKTk3n++ecpKvvD1dWV8ePHE  
xcXR0ZGBhUVFRQXF7Njxw5ef/111q5dS1BQEMePH7/s27plyxa2b99OeHg4ADNnziQgIICzZ8/StWtXh  
g4ditlsZuzYsaxevZrw8HCOHz+Ok5MTI0aMYNasWUyYMIGIS5diMpkIDg6+7GsKIYT4lbOTon/7RvS/z  
GImx4rP893eY6z+6RgnSkq5rVNjOjZtgI+7C6fPlXOs+Dyrfirgo+9ymLZ6P2AZhW7SwIOyCjPnyipo7u/JrI  
d61Gn8hkiSw4O86dUmyN5hCFFvLky5uJAkz5gxA4D//ve/TJs2jfLyco4cOcLOnTtrTZLXrFnD4MGD8fLy

AiApKcm2b/v27bz44oucPHmS4uLiKIM7apKdnU14eDjt2rUDYOTIkUyZMsWWJA8ZMgSAMJgY5s2bd  
9H5paW1LFq0iHfffRdfX1+6d+9OZmYmiYmJLF++nLS0NACcnZ3x8/MjLS2N4cOHExRk+TsPCAi47Hv  
WrVs3W4IMMHnyZDIyMgA4ePAge/bsoaCggH79+tmOu3DdMWPGkJyczIQJE5g5cyajR4++7OsJIYS4O  
kE+7iR3aU5yl+a1HjO6dzhFZ8tYt+8Y+wrOkHvsDL+cPo+7ixMers409/es87gMkSSP7h3O6N7hlz9QiGt1i  
RHf+pScnMzEiRPZsmULJSUlxEkJOTwzvvvMOmTZvw9/dn1KhRnDt37qquP2rUKObPn4/JZOLjjz9  
m5cqV1xSvu7vlLmpnZ2fKyy/+WiwzM5OTJ08SGRkJQEIJCZ6eniQmJv6m13FxccFsttwoYjabbbVNSALy  
9f32u58qVK1m6dCnr1q3Dy8uL+Pj4S75XoaGhNG7cmOXLI7Nx40ajL2MthBAOwc/TIYG3NL1urydzFIR  
wAD4+PvTv358xY8aQkpICwKITp/D29sbPz4/8/Hy+/vrrS16jX79+zJ8/n7Nnz3L69GkWLlxo23f69GmaN  
m1KWVIZIYTQ19eX06dPX3St9u3bk5uby969ewH49NNPiYuLu+L6pKenM336dHJzc8nNzSUnJ4cls5ZQ  
UIJCQkKC7akdFRUVFBUVceuttzJnzhwKCwsBbNMtwslC2Lx5MwALFingrKysxcrKirC398fLy8vdu/ez  
fr16wHo0aMHq1evJicnp8p1AR566CFGjBjB8OHDcXZ2vLuyhRBCXBtJkoVwECKpKXz//fe2JNlkMhEV  
FUWHDh2477776N279yXPj46O5p577sFkMnHnnXfStWtX276//vWvdO/end69e9OhQwdb+b333svbb79N  
VFQU+/bts5V7eHjw0UcfMXz4cCIjI3FycuLRRx+9onqUIJSwePFiBg0aZCvz9vamT58+LFy4kEmTJrFixQ  
oiIyOJiYlh586dRERE8MILLxAXF4fJZOKpp54CYOzYsaxatQqTycS6deuqjB5XNnDgQMrLy+nYsSOpq  
n06GGZtxYcHMy0adMYMmQIJpOJe+65x3ZOUlISxcXFMtVCCCFuUkprbe8YqoiNjdVZWVn2DkMI  
27dtGxY0d7hyGus6ysLCZOnMiaNWtqPaamz4ZSarPWOr+47uRSL8thHBUI+qzDTEnWQgh6tKbb77J1  
KITZS6yEELcxGS6hRBCVJOamsqBAwfo06ePvUMRQghhJ5IkCyGEEEIIUY0kyUJcgRtt7r6wP/IMCCG  
EsUmSLMRleHh4UFhYKEmRsNFaU1hYiIeHh71DEUIIUU/kxj0hLiMkJIS8vDwKCgrsHYq4gXh4eBAS  
EmLvMIQQQtQTSZKFuAxXV9cqyxsLIYQQwvhkuoUQQgghhBDVSJlshBBCCCFENZIkCyGEEEIIUc0  
Ntyy1UqoAOHAVpwYBx+o4nBuJ1M+xSf0c22+pX0utdXB9BnOjucp+Wz4zjk3q59ikfr+qtc++4ZLkq6W  
Uyqpt7W0jkPo5NqmfYzN6/ezB6O+p1M+xSf0cW13VT6ZbCCGEEEIIUY0kyUIIYQQQIRjpCR5mr0Dq  
GdSP8cm9XNsRq+fPRj9PZX6OTapn2Ork/oZZk6yEEIIIYQQdcVII8ICCCGEEELUCUMkyUqpgUqpbK  
XUXqVUqr3juRZKqVCI1Aql1E6l1A6l1B+t5QFKqSVKqt3W//rbO9ZroZRyVkpVUp9ad0OV0ptsLbh50  
opN3vHeLWUUg2VUnOVUruVUruUUj2N1H5KqYnWz+Z2pVS6UsrDkdtPKTVTKfWLUp7pbIa20tZ  
TLbW8welVLT9IndcRuqz4ebot43cZ4Ox+22j9dlw/fpth0+SIVL0wBTgTqATkKKU6mTfqK5JOfC01roT0  
AN43FqfVGcz1rotsMy67cj+COyqtP134B9a6zbACeAPdomqbkwCFmutOwAmLPU0RPspZoD44FYrfU  
tgDNwL47dfh8DA6uV1dZedwJtrT8PA1OvU4yGYcA+G26OftvIfTYYtN82aJ8N16vf1lo79A/QE8istP0c8J  
y946rD+v0PuA3IBppay5oC2faO7RrqFGL9AN8KfAkoLA/9dqmpTR3pB/ADcrDO969Uboj2A5oDB4EAw  
MXafnc4evsBYcD2y7UX8AGQUtNx8nPf77Wh+2xrnQzVbxu5z7bGb9h+26h9tjXueu+3HX4kmV8/ABfk  
WcscnlIqDIgCNgCNtdZHRluOAo3tFVcdeA94FjBbtwOBk1rrcuu2I7dhOFAAfGT9anK6Usobg7Sf1voQ8  
A7wM3AEKAI2Y5z2u6C29jjsf3MdGfo9NGi//R7G7bPBwP32TdRnQz3020ZIkg1JKeUDfAFM0FqfqrxP  
W/4p5JCPJVFKJQK/aK032zuWeuICRANTtdZRwBmqfUXn4O3nDyRj+Z9KM8Cbi7/yMhRHbi9xfRmx3  
74J+mwwcL99M/bZUHftZYQk+RAQWmk7xFrmsJRSrlg62lla63nW4nylVFPr/qbAL/aK7xr1BpKUUrAb  
Cxf300CGiqIXKzHOHib5gF5WusN1u25WDpfo7TfACBHa12gtS4D5mFpU6O03wW1tZfh+hs7MOR7aO  
B+2+h9Nhi7375Z+myoh37bCEnyJqCt9U5NNywT0hfYOaarppRSwAxgl9b63Uq7FgAjrb+PxDLnzeForZ/  
TWodorcOwtNVyrfX9wApgrPUwR67fUeCgUqq9tSgB2IIB2g/LV3Y9IFJe1s/qhfoZov0qqa29FgAPWu+  
W7gEUVfp6T1wZQ/XZYOx+2+h9Nhi+375Z+myoj37b3hOv62jy9l3AT8A+4AV7x3ONdemD5SuCH4Bt  
1p+7sMwBWwbsAZYCAfaOtQ7qGg98af29FbAR2AvMAdztHd811KsLkGVtw/mAv5HaD/gLsBvYDnw  
KuDty+wHpWObqlWEZUfpDbe2F5YalKda+5kcsd4zbvQ6O9mOkPttan5ui3zZqn22tj2H7baP12dY6XZd+  
W1bcE0IIIYQQohojTlcQQgghBCiTkmSLIQQQgghRDWSJAhhBBCCFGNJMICCCGEEJU8/8B4fho5HxS5n0AAAAASUVORK5C  
YII=\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": "2021-12-21T16:53:55.277140Z"
        }
    }
}
```

```
"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.execute_result": "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",

" 1,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\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",

" 1,\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",

" 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",

" 1,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\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",

" 0,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 1,\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",

" 1,\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",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 0,\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",

" 1,\n",

" 1,\n",

" 0,\n",

" 1,\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",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\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",

```

    " ...]"  

]  

},  

"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",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\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",

" 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",

" 1,\n",

" 1,\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",

" 1,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 1,\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",

" 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",

" 1,\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",

" 1,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 1,\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",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\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",

" 1,\n",

" 1,\n",

" 0,\n",

" 0,\n",

" 1,\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",

" 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",

" 1,\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",
    "...]"
]

},
"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": "628fbfcfd",  
  "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",  
      "start_time": "2021-12-21T16:54:03.041018Z"  
    }  
  }  
}
```

```
"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": {  
" " i m a g e / p n g " :  
"iVBORw0KGgoAAAANSUhEUgAAATkAAAE9CAYAAABwcBXnAAAAOXRFWHRTb2Z0d2FyZQB  
NYXRwbG90bGliIHZlcNpb24zLjUuMCwgaHR0cHM6Ly9tYXRwbG90bGliLm9yZy8/fFQqAAAACXB  
IWXMAAAsTAAALEwEAmpwYAAAS40lEQVR4nO3debhd872A8ffrRCKIUqEqhhgyiBAkgmjV1FTU  
XCqh7U21FG1pFeUaUjQ1Hv10tuE8IBKzUOCGtJeIYalqHloEA0JMmicpC1Jzu/+cXbSk9Eue521z6/v53n  
Ok73XWmev7+Y877PWHiOlhCTlaqWyB5CkIhk5SVkzcpKyZuQkZc3IScqakZOUtXZID9BStOuYon2nss  
dQndp2i43KHkF16o03JjNjxoxY1rr6ilz7TnTo8dWyx1CdGvf4pWWPoDq18w79lrvO01VJWTNykrJm5CR  
lzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTIJWTNy  
krJm5CRLzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOUtaMnKSsG  
TIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOUta  
MnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOUtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5S1dmU  
P8O9kxLDDGbRLb6bPaqTfIT8D4LTv7M0RBw1g+ntzABh26Z3c+/AL9NtyYy49YwgAETB8xN3c+cdn  
APj+4bsx9MABpJR4ftJUjh2LR98OL+cO6VW0WPzrnRavRMNDQ20a9eOcY9PWLTu4v++iFNPPpEp0  
6bTuXPnEquesT4VGLiL2An4BNAC/TimdV+T+6t01ox5jxA0P8utzvrHY8kuu/SMXXzNmsWXPvzqVnQ+/  
gAULmliv8xo8fsOp3DX2OT7z6U4cO+QLbPuV4fzjg3lce/4RHPKlvlw76vHWvCswe8f+ONSEZsyZQpj7  
r+PDTfaqKSp6l9hp6sR0QD8EhgE9AKGRESvovbXFoyb+CqzZv+tqm3//o95LFjQBECH9iuTUlq0rl1DAx  
07rExDw0p0XKU906bPLmRe1b+TT/whw8+9gIgoe5S6VeRjcv2BSSml11JKHwK/A/YvcH9t1tGDd2H8D  
acyYtjhrNmp46L12/femCdvPo0JN/0nxw3/HQsWNDF1+mwu/s0YXrnnHF6/fzjvz/k7Yx57qcTp1Roign0H  
DWRA/75ccflAIy68w7WX78LW/fpU/J09a3IyHUBprS4/mZlmVq4/KaH6LXvT9hh8Hm8PeN9zjvhoEXrn  
njuDfoePJzPfe0CTjpiIB3at2PNTh3ZZ9et2GKfYWw68DRW69iewXtvX+I9UGsY838P8+gTE7199D2M/N  
UvefihsVxw3s848ydnlz1a3Sv92dWIOCoiJkTEhDT/72WP0+rendVIU1MipcSVt46jX++Nl9rm5dffYc7fPm  
DLzddn9x16MnnqTGa8N4f585u4/Q9Ps2OfTUqYXK2pS5fm44N1112X/Q44kIfGPsgbk1+nf98+9Ni8K2+  
9+SY79d+Ot99+u+RJ60+RkXsL2LDF9Q0qyxaTUrospdQvpdQv2nVccnX21uu8xqLL++/ehxdenQbAxuuv  
TUND8/+ejT67Fj02WY83ps5kytuz6L/VJnRcZWUAduvfg5dff6f1B1ermTt3Lo2NjYsuP3D/ffTttz1/mfouL  
0+azMuTJtNlgw14dPxE1ltvvZKnrt9FPrv6BNAAtIjahOW6DgcMK3F/du/rcoXy+bzc6r7k6k35/DueMuJtd+  
nZj6x4bkFLijWmz+P5PrwdgwLabcuI3BzJv/gKamhLH/+wGZv51LjP/OpfbHniKR6/7MfMXNPH0S29yxS  
3jSr5nKtK777zDoQcfCMD8BfM5dPBhDPzSXivP1XZEy2ftan7jEXsDF9P8EpIrU0rDV7T9Suumzr0+G  
ph86hte++JS8seQXVq5x368eSTE5b5FHOhr5NLKd0N3F3kPiRpRUp/4kGSimTkJGXNyEnKmpGTlDUjJ  
ylrRk5S1oycpKwZOUIZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUIZM3KSsmbkJGXNyE  
nKmpGTlDUjJylrRk5S1oycpKwZOUIZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUIZM3K  
SsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUIZM3KSstZueSsiohFIC69W/k2VyymltEbBs0nSJ7bc  
yKWUOrXmIJUUhKpOVyPicxHxzcrLzhGxSbFjSVJtfGTkImIY8GPg1Mqi9sC1RQ4lSbVSzZHcgcb+wFy  
AINJUwFNZSW1CNZH7MKWUqDwJERGrFTuSJNVONZG7MSJGAmtGxJHAA8DIxY4lSbWx3GdXF0  
opXRgRXwTeB7oDZ6aU7i98MkmqgY+MXMWzQEeaT1mfLW4cSaqtap5d/TYwHjgIOBh4LCKOKHow  
SaqFao7kTgK2TSnNBIiItYFHgCuLHEySaqGaJx5mAo0trjdWlkls3VvRe1dPqFycBDweEXfQ/Jjc/sAzrTC  
bJH1iKzpdXfiC31crPwvdUdw4klRbK3qD/lmtOYgkFeEjn3iLiHWAk4EtgVUWLk8p7V7gXJJUE9U88fBb  
4CVgE+AsYDLwRIEzSVLNVBO5tVNKVwDzUkoPppSOADyKk9QmVPM6uXmVf6dFxJeBqcCnixtJk

mqnmsj9NCI+BfwIuARYA/hhoVNJUo1U8wb90ZWLS4Hdih1HkmprRS8GvoR/fpHNUIJKxxUykSTV0Iq  
O5Ca02hQVvbtyvOgxF7X2btVGzGz8oOwRVKfmNy33eGyFLwa+upBpJKkV+eXSkJm5CRLzchJylo1nw  
zcPSLGRMRzletbR8TpxY8mSZ9cNUdy19P8xdLzAFJKzwCDixxKkmqlmsitmlIav8Sy+UUMI0m1Vk3kZ  
kTEZvzzy6UPBqYVOpUk1Ug17139LnAZ0DMi3gJeB75W6FSSVCPVvHf1NWDPiFgNWCml1PhRvyNJ  
9aKaTwY+c4nrAKSUzi5oJkmqmWpOV+e2uLwKsA/wYjHjSFJtVXO6utg75iPiQuDewiaSpBr6OO94WB  
XYoNaDSFIRqnlM7ln++blyDcA6gI/HSWoTqnlMbp8Wl+cD76SUfDGwpDZhZGLiAbg3pRSz1aaR5Jqa  
oWPyaWUFgAvR8RGrTSPJNVUNaerawHPR8R4WrycJKW0X2FTSVKNVBO5MwqfQpIKUk3k9k4p/bjl  
gog4H3iwmJEkqXaqeZ3cF5exbFCtB5GkIqzoe1ePAY4FNo2IZ1qs6gSMK3owSaqFFZ2uXgfcA5wLnNJie  
WNKaVahU0lSjazoe1dnA7OBIA03jiTVlt/WJSlrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJ  
ylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyE  
nKmpGTlDUjJylrRk5S1oycpKwZOUlZM3KSsmbkJGXNyEnKmpGTlDUjJylrRk5S1oycpKwZOUlZM3K  
SsmbkJl1rSkcuv9A9thpG/YcsC1XjrwUgLvuuiU9B2xL184deeapJxdt/96smRy6/0C22Ghtzjj5ByVNrdbyo  
+8dRZ/uG7LHgO0WLXvhUWfYb+AX2GPnvgwdchCN778PwK03Xc/AXfov+tlw7Y48/+zTZY1edwqlX  
ERcGRHvRsRzRe2jLWtoaMfpZ5/PmEf/xO33juU3V4zglZdepHvPLRI59Q3sMOBzi23focMqnHjqME476  
7ySJIzrOuSwr3PtTXcutuyk44/h1GHnMGbck+z15f0Yccl/AXDQIUO4b+x47hs7nl+MuJKNNu7Kllv1KWP  
sulTkkdxVwf4F3n6b9pn1PstWfbYFYPPVondi8W0/emfYW3Xr0ZLNu3ZfaftXVVmP7HXemQ4cOrT2qSr  
DjgM+z5lprLbbstUl/ZscBnwdgl1334O5Rty/1e3fccgP7HXRIa4zYZhQWuZTSWGBWUbefkyl/mczzz/6Jbf  
r2L3sU1bHuPXtx792jABh9x61MnfrmUtuMuu1m9j/o0NYera75mFzJ5s6Zw9FDh3Dm8AvptMYaZY+jOn  
bRJSP5zRUjGbTbTsyZ08jKK7dfbP3ECeNZpeOq9Oy1ZUkT1qd2ZQ8QEUCBRwF02WDDkqdpXfPmzeP  
ooYM54ODBDNr3gLLHUZ3bvHsPrvv1LqD51HXM/b9fbP2dt97EAV/5ahmj1bXSj+RSSpellPqlIPp9eu11y  
h6n1aSUOPm477B5954ceezxZY+jNmDG9HcBaGpq4hcXncvXh3570bqmpiZG3XGLj8ctQ+lHcv+uJjz+C  
LfeeB09e/Vm0BeaH4s76fSz+fCDDxh2ygnMmjmdbw45kF69t+aam0cDsPM23WlsbGTevA+57+5RXHPza  
Lr33KLMu6GCfPfbX+fRcQ8xa+YM+m25GT865XTmzp3L1VeMAGDQPgdw6OH/sWj7xx55iPXX34CN  
u25a1sh1K1JKxdxwxPXAarkBn4B1gWErpihX9ztbb9E2j//BIIfo07WuIsidQvdp79wE8/dSTy/wLKexILqU0  
pKjblqRqlf6YnCQVychJypqRk5Q1Iycpa0ZOtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOu  
aMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iyc  
pa0ZOtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOtaMnKSsGTIJWTNykrJm5CRLzchJypq  
Rk5Q1Iycpa0ZOtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOtaMnKSsGTIJWTNykrJm5C  
RLzchJypqRk5Q1Iycpa0ZOtaMnKSsGTIJWTYuUUtkaLBIR04E3yp6jjnQGZpQ9hOqSfxuL2ziltM6yVtRV  
5LS4iJiQUupX9hyqP/5tVM/TVUIZM3KSsmbk6ttlZQ+guuXfRpV8TE5S1jySk5Q1I1eHImKviHg5IiZFxC  
llz6P6ERFXRsS7EfFc2bO0FUauzkREA/BLYBDQCxgSEb3KnUp15Cpgr7KHaEuMXP3pD0xKKb2WUv  
oQ+B2wf8kzqU6klMYCs8qeoy0xcvWnCzClxfU3K8skfQxGTlWjFz9eQvYsMX1DSrLJH0MRq7+PAF0  
i4hNIqI9MBi4s+SZpDbLyNWZINJ84HvAvcCLwI0ppefLnUr1IiKuBx4FekTEmxHxrbJnqne+40FS1jySk5  
Q1Iycpa0ZOtaMnKSsGTIJWTNykrJm5CRLzchJypqRk5Q1Iycpa0ZOtaMnKSsGTIJWTNykrJm5CRLzchJypq  
4KSNth5PSxVT4x5V+SUrozpXTeCjZZE/iXIyctj5HTUipHKi9FxG8j4sWIuDkiVq2smxwR50fEROCQiBg  
YEY9GxMSIuCkiVq9st1flNiYCB7W47aERcWnl8mci4raIeLryMwA4D9gsIv4UET+vbHdSRDwREc9ExF  
ktbuu0iHgllh4GelRvx46s3M7TEXHLwvtUsWdETKjc3j6V7Rsi4uct9v2dT/rfVq3PyGl5egD/m1LaAnifxY+  
uZqaUtgMeAE4H9qxcnwCcEBGrAJcD+wJ9gfWWs4//AR5MKfUBtgOeB04BXk0pbZNSOikiBglDaP4Iq  
m2AvhGxS0T0pfktb9sAewPbV3Gfbk0pbV/Z34tAy3cLdK3s48vAiMp9+BYwO6W0feX2j4yITarYj+pIu7I  
HUN2aklIaV718LXAccGHI+g2Vf3ek+YM9x0UEQHua33LUE3g9pfRngIi4FjhqGfvYHfgGQEppATA7ItZ  
aYpuBlZ+nKtdXpzl6nYDbUkp/q+yjmfv39o6In9J8SrW6zW+dW+jGIFT8OeIeK1yHwYCW7d4vO5TIX2/  
UsW+VCeMnJZnyff7tbw+t/JvAPenIa03DAitqnhHAGcm1IaucQ+fvAxbusq4ICU0tMRMRTYtcW6Zd3fA

L6fUmoZQyKi68fYt0ri6aqWZ6OI2Kly+TDg4WVs8xiwc0RsDhARq0VED+AloGtEbFbZbsgyfhdgDHBM  
5XcbIuJTQCPNR2kL3Qsc0eKxvi4RsS4wFjggIjpGRCeAT40/SidgWkSsDBy+xLpDImKlysbyAi9X9n1MZ  
XsiontErFbFflRHjJyW52XguxHxIrAW8KslN0gpTQeGAtdHxDNUTIVTSv+g+fT0rsoTD+8uZx/HA7tFxL  
PAk0CvlNJMMk9/n4uIn6eU7gOuAx6tbHcz0CmlNJHm0+angXto/oij3IG8DgwjuYQt/QXYHzlto6u3IdfA  
y8AEysvGRmJZz9tjp9CoqVUTsdGp5R6lz2L9E15JCcpax7JScqaR3KSsmbkJGXNyEnKmpGTIDUjJylrRk  
5S1v4fqFIJ4TT5IccAAAAASUVORK5CYII=\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
}
```