

```
1  {
2    "cells": [
3      {
4        "cell_type": "code",
5        "source": [
6          "import pandas as pd\n",
7          "import numpy as np"
8        ],
9        "metadata": {
10          "id": "SJ6tAkjj18NL"
11        },
12        "execution_count": 1,
13        "outputs": []
14      },
15      {
16        "cell_type": "code",
17        "source": [
18          "file=pd.read_csv(\"/Churn_Modelling.csv\")\n",
19          "pro=pd.DataFrame(file)\n",
20          "pro.head()"
21        ],
22        "metadata": {
23          "colab": {
24            "base_uri": "https://localhost:8080/",
25            "height": 270
26          },
27          "id": "v0z0pikV2AJU",
28          "outputId": "87b38785-c693-45e1-c40b-f6a23dfa4659"
29        },
30        "execution_count": 12,
31        "outputs": [
32          {
33            "output_type": "execute_result",
34            "data": {
35              "text/plain": [
```

```
36 " RowNumber CustomerId Surname CreditScore Geography Gender Age \\n",
37 "0 1 15634602 Hargrave 619 France Female 42 \n",
38 "1 2 15647311 Hill 608 Spain Female 41 \n",
39 "2 3 15619304 Onio 502 France Female 42 \n",
40 "3 4 15701354 Boni 699 France Female 39 \n",
41 "4 5 15737888 Mitchell 850 Spain Female 43 \n",
42 "\n",
43 " Tenure Balance NumOfProducts HasCrCard IsActiveMember \\n",
44 "0 2 0.00 1 1 1 \n",
45 "1 1 83807.86 1 0 1 \n",
46 "2 8 159660.80 3 1 0 \n",
47 "3 1 0.00 2 0 0 \n",
48 "4 2 125510.82 1 1 1 \n",
49 "\n",
50 " EstimatedSalary Exited \n",
51 "0 101348.88 1 \n",
52 "1 112542.58 0 \n",
53 "2 113931.57 1 \n",
54 "3 93826.63 0 \n",
55 "4 79084.10 0 "
56 ],
57 "text/html": [
58 "\n",
59 " <div id=\"df-02feae72-c761-4186-b470-318fcd07dce7\">\n",
60 " <div class=\"colab-df-container\">\n",
61 " <div>\n",
62 "<style scoped>\n",
63 " .dataframe tbody tr th:only-of-type {\n",
64 " vertical-align: middle;\n",
65 " }\n",
66 "\n",
67 " .dataframe tbody tr th {\n",
68 " vertical-align: top;\n",
```

```
69     "}\n",
70     "\n",
71     ".dataframe thead th {\n",
72     "    text-align: right;\n",
73     "}\n",
74     "</style>\n",
75     "<table border='1' class='dataframe'>\n",
76     "    <thead>\n",
77     "        <tr style='text-align: right;'\n",
78     "            <th></th>\n",
79     "            <th>RowNumber</th>\n",
80     "            <th>CustomerId</th>\n",
81     "            <th>Surname</th>\n",
82     "            <th>CreditScore</th>\n",
83     "            <th>Geography</th>\n",
84     "            <th>Gender</th>\n",
85     "            <th>Age</th>\n",
86     "            <th>Tenure</th>\n",
87     "            <th>Balance</th>\n",
88     "            <th>NumOfProducts</th>\n",
89     "            <th>HasCrCard</th>\n",
90     "            <th>IsActiveMember</th>\n",
91     "            <th>EstimatedSalary</th>\n",
92     "            <th>Exited</th>\n",
93     "        </tr>\n",
94     "    </thead>\n",
95     "    <tbody>\n",
96     "        <tr>\n",
97     "            <th>0</th>\n",
98     "            <td>1</td>\n",
99     "            <td>15634602</td>\n",
100     "            <td>Hargrave</td>\n",
101     "            <td>619</td>\n",
102     "            <td>France</td>\n",
103     "            <td>Female</td>\n",
104     "            <td>42</td>
```

```
105 " <td>2</td>\n",
106 " <td>0.00</td>\n",
107 " <td>1</td>\n",
108 " <td>1</td>\n",
109 " <td>1</td>\n",
110 " <td>101348.88</td>\n",
111 " <td>1</td>\n",
112 " </tr>\n",
113 " <tr>\n",
114 " <th>1</th>\n",
115 " <td>2</td>\n",
116 " <td>15647311</td>\n",
117 " <td>Hill</td>\n",
118 " <td>608</td>\n",
119 " <td>Spain</td>\n",
120 " <td>Female</td>\n",
121 " <td>41</td>\n",
122 " <td>1</td>\n",
123 " <td>83807.86</td>\n",
124 " <td>1</td>\n",
125 " <td>0</td>\n",
126 " <td>1</td>\n",
127 " <td>112542.58</td>\n",
128 " <td>0</td>\n",
129 " </tr>\n",
130 " <tr>\n",
131 " <th>2</th>\n",
132 " <td>3</td>\n",
133 " <td>15619304</td>\n",
134 " <td>Onio</td>\n",
135 " <td>502</td>\n",
136 " <td>France</td>\n",
137 " <td>Female</td>\n",
138 " <td>42</td>\n",
139 " <td>8</td>\n",
140 " <td>159660.80</td>\n",
```

141 " <td>3</td>\n",
142 " <td>1</td>\n",
143 " <td>0</td>\n",
144 " <td>113931.57</td>\n",
145 " <td>1</td>\n",
146 " </tr>\n",
147 " <tr>\n",
148 " <th>3</th>\n",
149 " <td>4</td>\n",
150 " <td>15701354</td>\n",
151 " <td>Bonni</td>\n",
152 " <td>699</td>\n",
153 " <td>France</td>\n",
154 " <td>Female</td>\n",
155 " <td>39</td>\n",
156 " <td>1</td>\n",
157 " <td>0.00</td>\n",
158 " <td>2</td>\n",
159 " <td>0</td>\n",
160 " <td>0</td>\n",
161 " <td>93826.63</td>\n",
162 " <td>0</td>\n",
163 " </tr>\n",
164 " <tr>\n",
165 " <th>4</th>\n",
166 " <td>5</td>\n",
167 " <td>15737888</td>\n",
168 " <td>Mitchell</td>\n",
169 " <td>850</td>\n",
170 " <td>Spain</td>\n",
171 " <td>Female</td>\n",
172 " <td>43</td>\n",
173 " <td>2</td>\n",
174 " <td>125510.82</td>\n",
175 " <td>1</td>\n",
176 " <td>1</td>\n".

```
177 "      <td>1</td>\n",
178 "      <td>79084.10</td>\n",
179 "      <td>0</td>\n",
180 "    </tr>\n",
181 "  </tbody>\n",
182 "</table>\n",
183 "</div>\n",
184 "    <button class=\"colab-df-convert\" onclick=\"convertToInteractive('df-02feae72-c761-4186-b470-318fcd07dce7')\" \n",
185 "      title=\"Convert this dataframe to an interactive table.\"\n",
186 "      style=\"display:none;\">\n",
187 "      \n",
188 "    <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\" viewBox=\"0 0 24 24\" \n",
189 "      width=\"24px\">\n",
190 "      <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",
191 "      <path d=\"M18.56 5.441 9.4 2.06 9.4-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-2.06 9.4zm-11 11 8.51 8.51 9.4-2.06 2.06-.94 2.06-.94 2.06-.94 2.06-2.06 9.4zm10\" \n",
192 "    </svg>\n",
193 "  </button>\n",
194 "  \n",
195 "  <style>\n",
196 "    .colab-df-container {\n",
197 "      display: flex;\n",
198 "      flex-wrap: wrap;\n",
199 "      gap: 12px;\n",
200 "    }\n",
201 "  \n",
202 "  .colab-df-convert {\n",
203 "    background-color: #E8F0FE;\n",
204 "    border: none;\n",
205 "    border-radius: 50%;\n",
206 "    cursor: pointer;\n",
207 "    display: none;\n",
208 "    fill: #1967D2;\n",
209 "    height: 32px;\n",
210 "    padding: 0 0 0 0;\n",
211 "    width: 32px;\n",
212 "  }\n",
```

```

213 "\n",
214 "  .colab-df-convert:hover {\n",
215 "    background-color: #E2EBFA;\n",
216 "    box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
217 "    fill: #174EA6;\n",
218 "  }\n",
219 "\n",
220 "  [theme=dark] .colab-df-convert {\n",
221 "    background-color: #3B4455;\n",
222 "    fill: #D2E3FC;\n",
223 "  }\n",
224 "\n",
225 "  [theme=dark] .colab-df-convert:hover {\n",
226 "    background-color: #434B5C;\n",
227 "    box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
228 "    filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
229 "    fill: #FFFFFF;\n",
230 "  }\n",
231 "</style>\n",
232 "\n",
233 "  <script>\n",
234 "    const buttonEl =\n",
235 "      document.querySelector('#df-02feae72-c761-4186-b470-318fcd07dce7 button.colab-df-convert');\n",
236 "    buttonEl.style.display =\n",
237 "      google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
238 "\n",
239 "    async function convertToInteractive(key) {\n",
240 "      const element = document.querySelector('#df-02feae72-c761-4186-b470-318fcd07dce7');\n",
241 "      const dataTable =\n",
242 "        await google.colab.kernel.invokeFunction('convertToInteractive',\n",
243 "          [key], {});\n",
244 "      if (!dataTable) return;\n",
245 "\n",
246 "      const docLinkHtml = 'Like what you see? Visit the ' +\n",
247 "        '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
248 "        + ' to learn more about interactive tables.';\n",

```

```

249         element.innerHTML = ''; \n",
250         dataTable['output_type'] = 'display_data'; \n",
251         await google.colab.output.renderOutput(dataTable, element); \n",
252         const docLink = document.createElement('div'); \n",
253         docLink.innerHTML = docLinkHtml; \n",
254         element.appendChild(docLink); \n",
255     } \n",
256     </script> \n",
257     </div> \n",
258     </div> \n",
259     "
260 ]
261 },
262 "metadata": {},
263 "execution_count": 12
264 }
265 ]
266 },
267 {
268     "cell_type": "code",
269     "source": [
270         "pro['HasCrCard'] = pro['HasCrCard'].astype('category')"
271     ],
272     "metadata": {
273         "id": "EdSjkjzx2gaG"
274     },
275     "execution_count": 13,
276     "outputs": []
277 },
278 {
279     "cell_type": "code",
280     "source": [
281         "pro['IsActiveMember'] = pro['IsActiveMember'].astype('category') \n",
282         "pro['Exited'] = pro['Exited'].astype('category') \n"
283     ],
284     "metadata": {

```



```

285         "id": "-JTPRKQy2jNE"
286     },
287     "execution_count": 14,
288     "outputs": []
289 },
290 {
291     "cell_type": "code",
292     "source": [
293         "\n",
294         "pro = pro.drop(columns=['RowNumber', 'CustomerId', 'Surname'])"
295     ],
296     "metadata": {
297         "id": "-5brBCAa2lmZ"
298     },
299     "execution_count": 15,
300     "outputs": []
301 },
302 {
303     "cell_type": "code",
304     "source": [
305         "pro.head()"
306     ],
307     "metadata": {
308         "colab": {
309             "base_uri": "https://localhost:8080/",
310             "height": 270
311         },
312         "id": "yo7Z3pwb2n_5",
313         "outputId": "15f9b900-27fc-46bd-808b-19b5c7cc0ee3"
314     },
315     "execution_count": 16,
316     "outputs": [
317         {
318             "output_type": "execute_result",
319             "data": {
320                 "text/plain": [

```

```

321 "    CreditScore Geography Gender Age Tenure Balance NumOfProducts  \\n",
322 "0        619    France Female  42    2      0.00          1  \\n",
323 "1        608    Spain  Female  41    1  83807.86          1  \\n",
324 "2        502    France Female  42    8 159660.80          3  \\n",
325 "3        699    France Female  39    1      0.00          2  \\n",
326 "4        850    Spain  Female  43    2 125510.82          1  \\n",
327 "\\n",
328 "    HasCrCard IsActiveMember EstimatedSalary Exited  \\n",
329 "0          1              1      101348.88      1  \\n",
330 "1          0              1      112542.58      0  \\n",
331 "2          1              0      113931.57      1  \\n",
332 "3          0              0       93826.63      0  \\n",
333 "4          1              1       79084.10      0  "
334 ],
335 "text/html": [
336 "\\n",
337 "    <div id=\"df-4f226628-a85d-41b8-80da-b9420a48fb14\">\\n",
338 "        <div class=\"colab-df-container\">\\n",
339 "            <div>\\n",
340 "                <style scoped>\\n",
341 "                    .dataframe tbody tr th:only-of-type {\\n",
342 "                        vertical-align: middle;\\n",
343 "                    }\\n",
344 "                \\n",
345 "                .dataframe tbody tr th {\\n",
346 "                    vertical-align: top;\\n",
347 "                }\\n",
348 "            \\n",
349 "            .dataframe thead th {\\n",
350 "                text-align: right;\\n",
351 "            }\\n",
352 "        </style>\\n",
353 "        <table border=\"1\" class=\"dataframe\">\\n",
354 "            <thead>\\n",
355 "                <tr style=\"text-align: right;\">\\n",
356 "                    <th></th>\\n",

```

```
357     <th>CreditScore</th>\n",
358     <th>Geography</th>\n",
359     <th>Gender</th>\n",
360     <th>Age</th>\n",
361     <th>Tenure</th>\n",
362     <th>Balance</th>\n",
363     <th>NumOfProducts</th>\n",
364     <th>HasCrCard</th>\n",
365     <th>IsActiveMember</th>\n",
366     <th>EstimatedSalary</th>\n",
367     <th>Exited</th>\n",
368   </tr>\n",
369 </thead>\n",
370 <tbody>\n",
371   <tr>\n",
372     <th>0</th>\n",
373     <td>619</td>\n",
374     <td>France</td>\n",
375     <td>Female</td>\n",
376     <td>42</td>\n",
377     <td>2</td>\n",
378     <td>0.00</td>\n",
379     <td>1</td>\n",
380     <td>1</td>\n",
381     <td>1</td>\n",
382     <td>101348.88</td>\n",
383     <td>1</td>\n",
384   </tr>\n",
385   <tr>\n",
386     <th>1</th>\n",
387     <td>608</td>\n",
388     <td>Spain</td>\n",
389     <td>Female</td>\n",
390     <td>41</td>\n",
391     <td>1</td>\n",
392     <td>83807.86</td>\n",
```

```
393 " <td>1</td>\n",
394 " <td>0</td>\n",
395 " <td>1</td>\n",
396 " <td>112542.58</td>\n",
397 " <td>0</td>\n",
398 " </tr>\n",
399 " <tr>\n",
400 " <th>2</th>\n",
401 " <td>502</td>\n",
402 " <td>France</td>\n",
403 " <td>Female</td>\n",
404 " <td>42</td>\n",
405 " <td>8</td>\n",
406 " <td>159660.80</td>\n",
407 " <td>3</td>\n",
408 " <td>1</td>\n",
409 " <td>0</td>\n",
410 " <td>113931.57</td>\n",
411 " <td>1</td>\n",
412 " </tr>\n",
413 " <tr>\n",
414 " <th>3</th>\n",
415 " <td>699</td>\n",
416 " <td>France</td>\n",
417 " <td>Female</td>\n",
418 " <td>39</td>\n",
419 " <td>1</td>\n",
420 " <td>0.00</td>\n",
421 " <td>2</td>\n",
422 " <td>0</td>\n",
423 " <td>0</td>\n",
424 " <td>93826.63</td>\n",
425 " <td>0</td>\n",
426 " </tr>\n",
427 " <tr>\n",
428 " <th>4</th>\n",
```

```

429         <td>850</td>\n",
430         <td>Spain</td>\n",
431         <td>Female</td>\n",
432         <td>43</td>\n",
433         <td>2</td>\n",
434         <td>125510.82</td>\n",
435         <td>1</td>\n",
436         <td>1</td>\n",
437         <td>1</td>\n",
438         <td>79084.10</td>\n",
439         <td>0</td>\n",
440     </tr>\n",
441 </tbody>\n",
442 </table>\n",
443 </div>\n",
444     <button class="colab-df-convert" onclick="convertToInteractive('df-4f226628-a85d-41b8-80da-b9420a48fb14')"\n",
445         title="Convert this dataframe to an interactive table.\n",
446         style="display:none;">\n",
447     \n",
448     <svg xmlns="http://www.w3.org/2000/svg" height="24px" viewBox="0 0 24 24"\n",
449     width="24px">\n",
450     <path d="M0 0h24v24H0V0z" fill="none"/>\n",
451     <path d="M18.5 5.44194 2.06 9.4-2.06-.94-2.06-.94-2.06-.94 2.06-2.06 9.4zm-11 11.5 8.51 9.4-2.06 2.06-.94-2.06-.94L8.5 2.51-.94 2.06-2.06.94
452 </svg>\n",
453 </button>\n",
454     \n",
455     <style>\n",
456     .colab-df-container {\n",
457     display: flex;\n",
458     flex-wrap: wrap;\n",
459     gap: 12px;\n",
460     }\n",
461     \n",
462     .colab-df-convert {\n",
463     background-color: #E8F0FE;\n",
464     border: none;\n",

```

```

465     "    border-radius: 50%;\n",
466     "    cursor: pointer;\n",
467     "    display: none;\n",
468     "    fill: #1967D2;\n",
469     "    height: 32px;\n",
470     "    padding: 0 0 0 0;\n",
471     "    width: 32px;\n",
472     "  }\n",
473 "\n",
474     "  .colab-df-convert:hover {\n",
475     "    background-color: #E2EBFA;\n",
476     "    box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
477     "    fill: #174EA6;\n",
478     "  }\n",
479 "\n",
480     "  [theme=dark] .colab-df-convert {\n",
481     "    background-color: #3B4455;\n",
482     "    fill: #D2E3FC;\n",
483     "  }\n",
484 "\n",
485     "  [theme=dark] .colab-df-convert:hover {\n",
486     "    background-color: #434B5C;\n",
487     "    box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
488     "    filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
489     "    fill: #FFFFFF;\n",
490     "  }\n",
491     "</style>\n",
492 "\n",
493     "<script>\n",
494     "    const buttonEl =\n",
495     "      document.querySelector('#df-4f226628-a85d-41b8-80da-b9420a48fb14 button.colab-df-convert');\n",
496     "    buttonEl.style.display =\n",
497     "      google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
498 "\n",
499     "    async function convertToInteractive(key) {\n",
500     "      const element = document.querySelector('#df-4f226628-a85d-41b8-80da-b9420a48fb14');\n",

```

```

501     "        const dataTable =\n",
502     "        await google.colab.kernel.invokeFunction('convertToInteractive',\n",
503     "        [key], {});\n",
504     "        if (!dataTable) return;\n",
505     "\n",
506     "        const docLinkHtml = 'Like what you see? Visit the ' +\n",
507     "        '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
508     "        + ' to learn more about interactive tables.';\n",
509     "        element.innerHTML = '';\n",
510     "        dataTable['output_type'] = 'display_data';\n",
511     "        await google.colab.output.renderOutput(dataTable, element);\n",
512     "        const docLink = document.createElement('div');\n",
513     "        docLink.innerHTML = docLinkHtml;\n",
514     "        element.appendChild(docLink);\n",
515     "    }\n",
516     "</script>\n",
517     "</div>\n",
518     "</div>\n",
519     " "
520 ]
521 },
522 "metadata": {},
523 "execution_count": 16
524 }
525 ]
526 },
527 {
528   "cell_type": "code",
529   "source": [
530     "import seaborn as sns\n",
531     "density = pro['Exited'].value_counts(normalize=True).reset_index()\n",
532     "sns.barplot(data=density, x='index', y='Exited', );\n",
533     "density"
534   ],
535   "metadata": {
536     "colab": {

```

```
537     "base_uri": "https://localhost:8080/",
538     "height": 374
539 },
540 "id": "y0D8vZlX2qpU",
541 "outputId": "c870051e-cf28-4377-d3f9-2e48bd9d0806"
542 },
543 "execution_count": 17,
544 "outputs": [
545   {
546     "output_type": "execute_result",
547     "data": {
548       "text/plain": [
549         "   index  Exited\n",
550         "0      0  0.7963\n",
551         "1      1  0.2037"
552       ],
553       "text/html": [
554         "\n",
555         "   <div id=\"df-80209c26-0c42-4011-8023-8f80bc381df8\">\n",
556         "     <div class=\"colab-df-container\">\n",
557         "       <div>\n",
558         "         <style scoped>\n",
559         "           .dataframe tbody tr th:only-of-type {\n",
560         "             vertical-align: middle;\n",
561         "           }\n",
562         "\n",
563         "           .dataframe tbody tr th {\n",
564         "             vertical-align: top;\n",
565         "           }\n",
566         "\n",
567         "           .dataframe thead th {\n",
568         "             text-align: right;\n",
569         "           }\n",
570         "</style>\n",
571         "<table border=\"1\" class=\"dataframe\">\n",
572         "   <thead>\n",
```



```

609     "}\n",
610 "\n",
611     ".colab-df-convert {\n",
612     "    background-color: #E8F0FE;\n",
613     "    border: none;\n",
614     "    border-radius: 50%;\n",
615     "    cursor: pointer;\n",
616     "    display: none;\n",
617     "    fill: #1967D2;\n",
618     "    height: 32px;\n",
619     "    padding: 0 0 0 0;\n",
620     "    width: 32px;\n",
621     "}\n",
622 "\n",
623     ".colab-df-convert:hover {\n",
624     "    background-color: #E2EBFA;\n",
625     "    box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
626     "    fill: #174EA6;\n",
627     "}\n",
628 "\n",
629     [theme=dark] .colab-df-convert {\n",
630     "    background-color: #3B4455;\n",
631     "    fill: #D2E3FC;\n",
632     "}\n",
633 "\n",
634     [theme=dark] .colab-df-convert:hover {\n",
635     "    background-color: #434B5C;\n",
636     "    box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
637     "    filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
638     "    fill: #FFFFFF;\n",
639     "}\n",
640     </style>\n",
641 "\n",
642     <script>\n",
643     "    const buttonEl =\n",
644     "        document.querySelector('#df-80209c26-0c42-4011-8023-8f80bc381df8 button.colab-df-convert');\n",

```

```

645         buttonEl.style.display =\n",
646         "         google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
647     "\n",
648     "         async function convertToInteractive(key) {\n",
649     "             const element = document.querySelector('#df-80209c26-0c42-4011-8023-8f80bc381df8');\n",
650     "             const dataTable =\n",
651     "                 await google.colab.kernel.invokeFunction('convertToInteractive',\n",
652     "                                                             [key], {});\n",
653     "             if (!dataTable) return;\n",
654     "\n",
655     "             const docLinkHtml = 'Like what you see? Visit the ' +\n",
656     "                 '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
657     "                 + ' to learn more about interactive tables.';\n",
658     "             element.innerHTML = '';\n",
659     "             dataTable['output_type'] = 'display_data';\n",
660     "             await google.colab.output.renderOutput(dataTable, element);\n",
661     "             const docLink = document.createElement('div');\n",
662     "             docLink.innerHTML = docLinkHtml;\n",
663     "             element.appendChild(docLink);\n",
664     "         }\n",
665     "     </script>\n",
666     " </div>\n",
667     " </div>\n",
668     " "
669 ]
670 },
671 "metadata": {},
672 "execution_count": 17
673 },
674 {
675     "output_type": "display_data",
676     "data": {
677         "text/plain": [
678             "<Figure size 432x288 with 1 Axes>"
679         ],
680         "image/png": "iVBORw0KGgoAAAANSUhEUgAAAYIAAAEGCAYAAABo25JHAAAABHNCSVQICAgIfAhkiAAAA1wSF1zAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsI"

```

```

681     },
682     "metadata": {
683         "needs_background": "light"
684     }
685 }
686 ]
687 },
688 {
689     "cell_type": "code",
690     "source": [
691         "import matplotlib.pyplot as plt"
692     ],
693     "metadata": {
694         "id": "ukLd8rcG2s_o"
695     },
696     "execution_count": 18,
697     "outputs": []
698 },
699 {
700     "cell_type": "code",
701     "source": [
702         "categorical = pro.drop(columns=['CreditScore', 'Age', 'Tenure', 'Balance', 'EstimatedSalary'])\n",
703         "rows = int(np.ceil(categorical.shape[1] / 2)) - 1\n",
704         "\n",
705         "# create sub-plots anf title them\n",
706         "fig, axes = plt.subplots(nrows=rows, ncols=2, figsize=(10,6))\n",
707         "axes = axes.flatten()\n",
708         "\n",
709         "for row in range(rows):\n",
710             cols = min(2, categorical.shape[1] - row*2)\n",
711             for col in range(cols):\n",
712                 col_name = categorical.columns[2 * row + col]\n",
713                 ax = axes[row*2 + col]\n",
714                 \n",
715                 sns.countplot(data=categorical, x=col_name, hue="Exited", ax=ax)\n",
716                 \n",

```

```
717         "plt.tight_layout()"
718     ],
719     "metadata": {
720         "colab": {
721             "base_uri": "https://localhost:8080/",
722             "height": 441
723         },
724         "id": "pjZgJrab2vd_",
725         "outputId": "e04a5991-d246-42c2-ec67-235fff0b664e"
726     },
727     "execution_count": 22,
728     "outputs": [
729         {
730             "output_type": "display_data",
731             "data": {
732                 "text/plain": [
733                     "<Figure size 720x432 with 4 Axes>"
734                 ],
735                 "image/png": "iVBORw0KGgoAAAANSUgAAAsgAAAGoCAYAAABbtX0xAAAABHNCSVQICAgIFAhkiAAAAA1wSF1zAAALEgAACxIB0t1+/AAADh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsI
736             },
737             "metadata": {
738                 "needs_background": "light"
739             }
740         }
741     ]
742 },
743 {
744     "cell_type": "code",
745     "source": [
746         "pro.info"
747     ],
748     "metadata": {
749         "colab": {
750             "base_uri": "https://localhost:8080/"
751         },
752         "id": "215Kd1T2xbD",
```

```

753     "outputId": "6b907ba3-bd99-488a-c175-d424babfc174"
754 },
755 "execution_count": 23,
756 "outputs": [
757   {
758     "output_type": "execute_result",
759     "data": {
760       "text/plain": [
761         "<bound method DataFrame.info of          CreditScore Geography Gender Age  Tenure   Balance  NumOfProducts  \\n",
762         "0          619   France  Female  42      2    0.00          1  \n",
763         "1          608   Spain  Female  41      1  83807.86          1  \n",
764         "2          502   France  Female  42      8  159660.80          3  \n",
765         "3          699   France  Female  39      1    0.00          2  \n",
766         "4          850   Spain  Female  43      2  125510.82          1  \n",
767         "...      ...      ...      ...      ...      ...      ...      ...  \n",
768         "9995         771   France   Male  39      5    0.00          2  \n",
769         "9996         516   France   Male  35     10  57369.61          1  \n",
770         "9997         709   France  Female  36      7    0.00          1  \n",
771         "9998         772  Germany   Male  42      3  75075.31          2  \n",
772         "9999         792   France  Female  28      4  130142.79          1  \n",
773         "\n",
774         "      HasCrCard IsActiveMember  EstimatedSalary Exited  \n",
775         "0              1              1      101348.88      1  \n",
776         "1              0              1      112542.58      0  \n",
777         "2              1              0      113931.57      1  \n",
778         "3              0              0       93826.63      0  \n",
779         "4              1              1       79084.10      0  \n",
780         "...      ...      ...      ...      ...  \n",
781         "9995          1              0       96270.64      0  \n",
782         "9996          1              1      101699.77      0  \n",
783         "9997          0              1       42085.58      1  \n",
784         "9998          1              0       92888.52      1  \n",
785         "9999          1              0       38190.78      0  \n",
786         "\n",
787         "[10000 rows x 11 columns]>"
788       ]

```

```

789     },
790     "metadata": {},
791     "execution_count": 23
792 }
793 ]
794 },
795 {
796     "cell_type": "code",
797     "source": [
798         "pro.isna().sum()"
799     ],
800     "metadata": {
801         "colab": {
802             "base_uri": "https://localhost:8080/"
803         },
804         "id": "g9oYJ3013n0m",
805         "outputId": "48d62ed5-01eb-4448-c688-4ce52dfc82c0"
806     },
807     "execution_count": 25,
808     "outputs": [
809         {
810             "output_type": "execute_result",
811             "data": {
812                 "text/plain": [
813                     "CreditScore      0\n",
814                     "Geography      0\n",
815                     "Gender      0\n",
816                     "Age      0\n",
817                     "Tenure      0\n",
818                     "Balance      0\n",
819                     "NumOfProducts  0\n",
820                     "HasCrCard      0\n",
821                     "IsActiveMember 0\n",
822                     "EstimatedSalary 0\n",
823                     "Exited      0\n",
824                     "dtype: int64"

```

```

825         ]
826     },
827     "metadata": {},
828     "execution_count": 25
829 }
830 ]
831 },
832 {
833     "cell_type": "code",
834     "source": [
835         "for i in pro:\n",
836         "    if pro[i].dtype=='object' or pro[i].dtype=='category':\n",
837         "        print(\"unique of \"+i+"\n is \""+str(len(set(df[i])))+"\n they are \""+str(set(df[i])))"
838     ],
839     "metadata": {
840         "colab": {
841             "base_uri": "https://localhost:8080/"
842         },
843         "id": "19WMTxRb3tCt",
844         "outputId": "132fd37c-1aaa-4c09-f3fa-8b64803f97cb"
845     },
846     "execution_count": 26,
847     "outputs": [
848         {
849             "output_type": "stream",
850             "name": "stdout",
851             "text": [
852                 "unique of Geography is 3 they are {'Germany', 'France', 'Spain'}\n",
853                 "unique of Gender is 2 they are {'Male', 'Female'}\n",
854                 "unique of HasCrCard is 2 they are {0, 1}\n",
855                 "unique of IsActiveMember is 2 they are {0, 1}\n",
856                 "unique of Exited is 2 they are {0, 1}\n"
857             ]
858         }
859     ]
860 },

```



```

861 {
862     "cell_type": "code",
863     "source": [
864         "def box_scatter(data, x, y):    \n",
865         "    fig, (ax1, ax2) = plt.subplots(nrows=2, ncols=1, figsize=(16,6))\n",
866         "    sns.boxplot(data=data, x=x, ax=ax1)\n",
867         "    sns.scatterplot(data=data, x=x,y=y,ax=ax2)"
868     ],
869     "metadata": {
870         "id": "xdD0mq0L3xdJ"
871     },
872     "execution_count": 27,
873     "outputs": []
874 },
875 {
876     "cell_type": "code",
877     "source": [
878         "box_scatter(pro,'CreditScore','Exited');\n",
879         "plt.tight_layout()\n",
880         "print(f\"# of Bivariate Outliers: {len(pro.loc[pro['CreditScore'] < 400])})\")"
881     ],
882     "metadata": {
883         "colab": {
884             "base_uri": "https://localhost:8080/",
885             "height": 375
886         },
887         "id": "bG8cCm1D32ij",
888         "outputId": "e2fc30c4-eea6-4350-a226-7780971cb304"
889     },
890     "execution_count": 28,
891     "outputs": [
892         {
893             "output_type": "stream",
894             "name": "stdout",
895             "text": [
896                 "# of Bivariate Outliers: 19\n"

```

```
897     ]
898 },
899 {
900     "output_type": "display_data",
901     "data": {
902         "text/plain": [
903             "<Figure size 1152x432 with 2 Axes>"
904         ],
905         "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLAAAA BHNCVQICAgIFAhkiAAAAA1wSF1zAAALEgAACxIB0t1+/AAAA Dh0RVh0U29mdHdhcmUAAbWF0cGxwdGxpYiB2ZXJzaW9uMy4yLjIs
906     },
907     "metadata": {
908         "needs_background": "light"
909     }
910 }
911 ]
912 },
913 {
914     "cell_type": "code",
915     "source": [
916         "box_scatter(pro, 'Age', 'Exited');\n",
917         "plt.tight_layout()\n",
918         "print(f'\# of Bivariate Outliers: {len(pro.loc[pro['Age'] > 87])}')\n"
919     ],
920     "metadata": {
921         "colab": {
922             "base_uri": "https://localhost:8080/",
923             "height": 375
924         },
925         "id": "prJ9bPI3380g",
926         "outputId": "f93eb05f-575e-4511-b09c-873a130eab33"
927     },
928     "execution_count": 29,
929     "outputs": [
930         {
931             "output_type": "stream",
932             "name": "stdout",
```

```
933     "text": [
934         "# of Bivariate Outliers: 3\n"
935     ]
936 },
937 {
938     "output_type": "display_data",
939     "data": {
940         "text/plain": [
941             "<Figure size 1152x432 with 2 Axes>"
942         ],
943         "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLAAAAABHNCSVQICAgIfAhkiAAAAA1wSF1zAAALEgAACxIB0t1+/AAADh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsI
944     },
945     "metadata": {
946         "needs_background": "light"
947     }
948 }
949 ]
950 },
951 {
952     "cell_type": "code",
953     "source": [
954         "box_scatter(pro,'Balance','Exited');\n",
955         "plt.tight_layout()\n",
956         "print(f\"# of Bivariate Outliers: {len(pro.loc[pro['Balance'] > 220000])}\")"
957     ],
958     "metadata": {
959         "colab": {
960             "base_uri": "https://localhost:8080/",
961             "height": 375
962         },
963         "id": "TGM0sMF84Biu",
964         "outputId": "548c050b-616e-41a3-c977-60b0a1112943"
965     },
966     "execution_count": 30,
967     "outputs": [
968         {
```

```
969         "output_type": "stream",
970         "name": "stdout",
971         "text": [
972             "# of Bivariate Outliers: 4\n"
973         ]
974     },
975     {
976         "output_type": "display_data",
977         "data": {
978             "text/plain": [
979                 "<Figure size 1152x432 with 2 Axes>"
980             ],
981             "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLAAAAABHNCSVQICAgIFAhkiAAAAAlwSFlzAAALEgAACxIB0t1+/AAAADh0RVh0U29mdHdhcmUAJabWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIs
982         },
983         "metadata": {
984             "needs_background": "light"
985         }
986     }
987 ]
988 },
989 {
990     "cell_type": "code",
991     "source": [
992         "box_scatter(pro, 'EstimatedSalary', 'Exited');\n",
993         "plt.tight_layout()"
994     ],
995     "metadata": {
996         "colab": {
997             "base_uri": "https://localhost:8080/",
998             "height": 357
999         },
1000         "id": "Pfob2x0V4G3o",
1001         "outputId": "3ebc8545-40de-4107-bd6d-d488754af9d9"
1002     },
1003     "execution_count": 31,
1004     "outputs": [
```

```
1005     {
1006         "output_type": "display_data",
1007         "data": {
1008             "text/plain": [
1009                 "<Figure size 1152x432 with 2 Axes>"
1010             ],
1011             "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLAAAAABHNCSVQICAgIfAhkiAAAAA1wSF1zAAALEgAACxIB0t1+/AAAAh0RVh0U29mdHdhcmUAbWl0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsI
1012         },
1013         "metadata": {
1014             "needs_background": "light"
1015         }
1016     }
1017 ]
1018 },
1019 {
1020     "cell_type": "code",
1021     "source": [
1022         "for i in pro:\n",
1023         "    if pro[i].dtype=='int64' or pro[i].dtypes=='float64':\n",
1024         "        q1=pro[i].quantile(0.25)\n",
1025         "        q3=pro[i].quantile(0.75)\n",
1026         "        iqr=q3-q1\n",
1027         "        upper=q3+1.5*iqr\n",
1028         "        lower=q1-1.5*iqr\n",
1029         "        pro[i]=np.where(pro[i] >upper, upper, pro[i])\n",
1030         "        pro[i]=np.where(pro[i] <lower, lower, pro[i])\n",
1031         "
1032     ],
1033     "metadata": {
1034         "id": "FLL1a9bX4KuP"
1035     },
1036     "execution_count": 32,
1037     "outputs": []
1038 },
1039 {
1040     "cell_type": "code",
```

```
1041 "source": [  
1042     "box_scatter(pro, 'CreditScore', 'Exited');\n",  
1043     "plt.tight_layout()\n",  
1044     "print(f'\n# of Bivariate Outliers: {len(pro.loc[pro['CreditScore'] < 400])}\n")"  
1045 ],  
1046 "metadata": {  
1047     "colab": {  
1048         "base_uri": "https://localhost:8080/",  
1049         "height": 375  
1050     },  
1051     "id": "poNE_CUo4U2m",  
1052     "outputId": "586932d9-e3c9-40cb-cd28-4cd6f3682315"  
1053 },  
1054 "execution_count": 33,  
1055 "outputs": [  
1056     {  
1057         "output_type": "stream",  
1058         "name": "stdout",  
1059         "text": [  
1060             "# of Bivariate Outliers: 19\n"  
1061         ]  
1062     },  
1063     {  
1064         "output_type": "display_data",  
1065         "data": {  
1066             "text/plain": [  
1067                 "<Figure size 1152x432 with 2 Axes>"  
1068             ],  
1069             "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLAAABHNCSVQICAgIfAhkiAAAA1wSF1zAAAEgAACxIB0t1+/AAAAh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsI  
1070         ],  
1071         "metadata": {  
1072             "needs_background": "light"  
1073         }  
1074     }  
1075 ]  
1076 },
```

```

1077 {
1078     "cell_type": "code",
1079     "source": [
1080         "box_scatter(pro, 'Age', 'Exited');\n",
1081         "plt.tight_layout()\n",
1082         "print(f'\n# of Bivariate Outliers: {len(pro.loc[pro['Age'] > 87])}\n')\"
1083     ],
1084     "metadata": {
1085         "colab": {
1086             "base_uri": "https://localhost:8080/",
1087             "height": 375
1088         },
1089         "id": "OJA8dEwz4aKx",
1090         "outputId": "1b69d2f7-f974-425a-b8f9-2fe6bec22f1a"
1091     },
1092     "execution_count": 34,
1093     "outputs": [
1094         {
1095             "output_type": "stream",
1096             "name": "stdout",
1097             "text": [
1098                 "\n# of Bivariate Outliers: 0\n"
1099             ]
1100         },
1101         {
1102             "output_type": "display_data",
1103             "data": {
1104                 "text/plain": [
1105                     "<Figure size 1152x432 with 2 Axes>"
1106                 ],
1107                 "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLLAAAAABHNCSVQICAgIfAhkiAAAAA1wSF1zAAALEgAACxIB0t1+/AAAA Dh0RVh0U29mdHdhcmUAAbWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsI
1108             },
1109             "metadata": {
1110                 "needs_background": "light"
1111             }
1112         }
1113     ]
1114 }

```

```
1113     ]
1114 },
1115 {
1116     "cell_type": "code",
1117     "source": [
1118         "box_scatter(pro,'Balance','Exited');\n",
1119         "plt.tight_layout()\n",
1120         "print(f\"# of Bivariate Outliers: {len(pro.loc[pro['Balance'] > 220000])}\")"
1121     ],
1122     "metadata": {
1123         "colab": {
1124             "base_uri": "https://localhost:8080/",
1125             "height": 375
1126         },
1127         "id": "3r0ST9_j4fy5",
1128         "outputId": "e7a45806-7784-4de4-bd3d-2ff4fc6acc8d"
1129     },
1130     "execution_count": 35,
1131     "outputs": [
1132         {
1133             "output_type": "stream",
1134             "name": "stdout",
1135             "text": [
1136                 "# of Bivariate Outliers: 4\n"
1137             ]
1138         },
1139         {
1140             "output_type": "display_data",
1141             "data": {
1142                 "text/plain": [
1143                     "<Figure size 1152x432 with 2 Axes>"
1144                 ],
1145                 "image/png": "iVBORw0KGgoAAAANSUhEUgAABHgAAAGoCAYAAAAA99FLAAABHNCSVQICAgIFAhkiAAAAA1wSF1zAAALEgAACxIB0t1+/AAAAOh0RVh0U29mdHdhcmUAblWF0cGxvdGxpYiB2ZXJzaW9uMy4yLjIsI"
1146             },
1147             "metadata": {
1148                 "needs_background": "light"
1149             }
1150         }
1151     ]
1152 }
```



```

1149     }
1150 }
1151 ]
1152 },
1153 {
1154     "cell_type": "code",
1155     "source": [
1156         "from sklearn.preprocessing import LabelEncoder\n",
1157         "encoder=LabelEncoder()\n",
1158         "for i in pro:\n",
1159         "    if pro[i].dtype=='object' or pro[i].dtype=='category':\n",
1160         "        pro[i]=encoder.fit_transform(pro[i])"
1161     ],
1162     "metadata": {
1163         "id": "jg4CVChY419s"
1164     },
1165     "execution_count": 36,
1166     "outputs": []
1167 },
1168 {
1169     "cell_type": "code",
1170     "source": [
1171         "x=pro.iloc[:, :-1]\n",
1172         "x.head()"
1173     ],
1174     "metadata": {
1175         "colab": {
1176             "base_uri": "https://localhost:8080/",
1177             "height": 270
1178         },
1179         "id": "ciUAjg9L4sL1",
1180         "outputId": "706fca47-aba7-4a8e-b7c5-ddf13ea36f89"
1181     },
1182     "execution_count": 37,
1183     "outputs": [
1184         {

```

```
1185 "output_type": "execute_result",
1186 "data": {
1187   "text/plain": [
1188     "   CreditScore  Geography  Gender  Age  Tenure   Balance  NumOfProducts  \\n",
1189     "0          619.0         0      0  42.0    2.0     0.00         1.0  \\n",
1190     "1          608.0         2      0  41.0    1.0    83807.86         1.0  \\n",
1191     "2          502.0         0      0  42.0    8.0   159660.80         3.0  \\n",
1192     "3          699.0         0      0  39.0    1.0     0.00         2.0  \\n",
1193     "4          850.0         2      0  43.0    2.0   125510.82         1.0  \\n",
1194     "\\n",
1195     "   HasCrCard  IsActiveMember  EstimatedSalary  \\n",
1196     "0           1                1         101348.88  \\n",
1197     "1           0                1         112542.58  \\n",
1198     "2           1                0         113931.57  \\n",
1199     "3           0                0          93826.63  \\n",
1200     "4           1                1          79084.10  "
1201   ],
1202   "text/html": [
1203     "\\n",
1204     "   <div id=\"df-71330b13-2873-41c2-9dc0-b1458361a20d\">\\n",
1205     "     <div class=\"colab-df-container\">\\n",
1206     "       <div>\\n",
1207     "         <style scoped>\\n",
1208     "           .dataframe tbody tr th:only-of-type {\\n",
1209     "             vertical-align: middle;\\n",
1210     "           }\\n",
1211     "         \\n",
1212     "           .dataframe tbody tr th {\\n",
1213     "             vertical-align: top;\\n",
1214     "           }\\n",
1215     "         \\n",
1216     "           .dataframe thead th {\\n",
1217     "             text-align: right;\\n",
1218     "           }\\n",
1219     "       </style>\\n",
1220     "     <table border=\"1\" class=\"dataframe\">\\n",
```

```
1221 " <thead>\n",
1222 " <tr style=\"text-align: right;\">\n",
1223 " <th></th>\n",
1224 " <th>CreditScore</th>\n",
1225 " <th>Geography</th>\n",
1226 " <th>Gender</th>\n",
1227 " <th>Age</th>\n",
1228 " <th>Tenure</th>\n",
1229 " <th>Balance</th>\n",
1230 " <th>NumOfProducts</th>\n",
1231 " <th>HasCrCard</th>\n",
1232 " <th>IsActiveMember</th>\n",
1233 " <th>EstimatedSalary</th>\n",
1234 " </tr>\n",
1235 " </thead>\n",
1236 " <tbody>\n",
1237 " <tr>\n",
1238 " <th>0</th>\n",
1239 " <td>619.0</td>\n",
1240 " <td>0</td>\n",
1241 " <td>0</td>\n",
1242 " <td>42.0</td>\n",
1243 " <td>2.0</td>\n",
1244 " <td>0.00</td>\n",
1245 " <td>1.0</td>\n",
1246 " <td>1</td>\n",
1247 " <td>1</td>\n",
1248 " <td>101348.88</td>\n",
1249 " </tr>\n",
1250 " <tr>\n",
1251 " <th>1</th>\n",
1252 " <td>608.0</td>\n",
1253 " <td>2</td>\n",
1254 " <td>0</td>\n",
1255 " <td>41.0</td>\n",
1256 " <td>1.0</td>
```

```
1257 "      <td>83807.86</td>\n",
1258 "      <td>1.0</td>\n",
1259 "      <td>0</td>\n",
1260 "      <td>1</td>\n",
1261 "      <td>112542.58</td>\n",
1262 "    </tr>\n",
1263 "  <tr>\n",
1264 "    <th>2</th>\n",
1265 "    <td>502.0</td>\n",
1266 "    <td>0</td>\n",
1267 "    <td>0</td>\n",
1268 "    <td>42.0</td>\n",
1269 "    <td>8.0</td>\n",
1270 "    <td>159660.80</td>\n",
1271 "    <td>3.0</td>\n",
1272 "    <td>1</td>\n",
1273 "    <td>0</td>\n",
1274 "    <td>113931.57</td>\n",
1275 "  </tr>\n",
1276 " <tr>\n",
1277 "   <th>3</th>\n",
1278 "   <td>699.0</td>\n",
1279 "   <td>0</td>\n",
1280 "   <td>0</td>\n",
1281 "   <td>39.0</td>\n",
1282 "   <td>1.0</td>\n",
1283 "   <td>0.00</td>\n",
1284 "   <td>2.0</td>\n",
1285 "   <td>0</td>\n",
1286 "   <td>0</td>\n",
1287 "   <td>93826.63</td>\n",
1288 " </tr>\n",
1289 " <tr>\n",
1290 "   <th>4</th>\n",
1291 "   <td>850.0</td>\n",
1292 "   <td>2</td>\n",
```

```
1293 "      <td>0</td>\n",  
1294 "      <td>43.0</td>\n",  
1295 "      <td>2.0</td>\n",  
1296 "      <td>125510.82</td>\n",  
1297 "      <td>1.0</td>\n",  
1298 "      <td>1</td>\n",  
1299 "      <td>1</td>\n",  
1300 "      <td>79084.10</td>\n",  
1301 "    </tr>\n",  
1302 "  </tbody>\n",  
1303 "</table>\n",  
1304 "</div>\n",  
1305 "    <button class=\"\`colab-df-convert\" onclick=\"\`convertToInteractive(df=71330b13-2873-41c2-9dc0-b1458361a20d')\"\`>\n",  
1306 "        title=\"Convert this dataframe to an interactive table.\"\\n\",  
1307 "        style=\"display:none;\">\n",  
1308 "          \n",  
1309 "    <svg xmlns=\"http://www.w3.org/2000/svg\" height=\"24px\" viewBox=\"0 0 24 24\"\\n",  
1310 "        width=\"24px\">\n",  
1311 "      <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",  
1312 "      <path d=\"M18.56 5.44l.94 2.06-.94 2.06-.94 2.06-.94 2.06-.94 2.06-.94zm-11 11L8.5 8.5l.94 2.06-.94 2.06-.94 2.06-.94zm10 -3.56l.94 2.06-.94 2.06-.94 2.06-.94zm-2.51 -.94 2.06-2.06.94zm10  
1313 "    </svg>\n",  
1314 "  </button>\n",  
1315 "  \n",  
1316 "<style>\n",  
1317 ".colab-df-container {\n",  
1318   display:flex;\n",  
1319   flex-wrap:wrap;\n",  
1320   gap: 12px;\n",  
1321 "}\n",  
1322 "\n",  
1323 ".colab-df-convert {\n",  
1324   background-color: #E8F0FE;\n",  
1325   border: none;\n",  
1326   border-radius: 50%;\n",  
1327   cursor: pointer;\n",  
1328   display: none;
```

```

1329 "    fill: #1967D2;\n",
1330 "    height: 32px;\n",
1331 "    padding: 0 0 0 0;\n",
1332 "    width: 32px;\n",
1333 "  }\n",
1334 "\n",
1335 "  .colab-df-convert:hover {\n",
1336 "    background-color: #E2EBFA;\n",
1337 "    box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px rgba(60, 64, 67, 0.15);\n",
1338 "    fill: #174EA6;\n",
1339 "  }\n",
1340 "\n",
1341 "  [theme=dark] .colab-df-convert {\n",
1342 "    background-color: #384455;\n",
1343 "    fill: #D2E3FC;\n",
1344 "  }\n",
1345 "\n",
1346 "  [theme=dark] .colab-df-convert:hover {\n",
1347 "    background-color: #434B5C;\n",
1348 "    box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
1349 "    filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
1350 "    fill: #FFFFFF;\n",
1351 "  }\n",
1352 "</style>\n",
1353 "\n",
1354 "<script>\n",
1355 "  const buttonEl =\n",
1356 "    document.querySelector('#df-71330b13-2873-41c2-9dc0-b1458361a20d button.colab-df-convert');\n",
1357 "  buttonEl.style.display =\n",
1358 "    google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
1359 "\n",
1360 "  async function convertToInteractive(key) {\n",
1361 "    const element = document.querySelector('#df-71330b13-2873-41c2-9dc0-b1458361a20d');\n",
1362 "    const dataTable =\n",
1363 "      await google.colab.kernel.invokeFunction('convertToInteractive',\n",
1364 "        [key], {});\n",

```

```

1365         "            if (!dataTable) return;\n",
1366         "\n",
1367         "            const docLinkHtml = 'Like what you see? Visit the ' +\n",
1368         "                '<a target=\"_blank\" href=https://colab.research.google.com/notebooks/data_table.ipynb>data table notebook</a>'\n",
1369         "                + ' to learn more about interactive tables.';\n",
1370         "            element.innerHTML = '';\n",
1371         "            dataTable['output_type'] = 'display_data';\n",
1372         "            await google.colab.output.renderOutput(dataTable, element);\n",
1373         "            const docLink = document.createElement('div');\n",
1374         "            docLink.innerHTML = docLinkHtml;\n",
1375         "            element.appendChild(docLink);\n",
1376         "        }\n",
1377         "</script>\n",
1378         "</div>\n",
1379         "</div>\n",
1380         "    "
1381     ]
1382 },
1383 "metadata": {},
1384 "execution_count": 37
1385 }
1386 ]
1387 },
1388 {
1389     "cell_type": "code",
1390     "source": [
1391         "y=pro.iloc[:,-1]\n",
1392         "y.head()"
1393     ],
1394     "metadata": {
1395         "colab": {
1396             "base_uri": "https://localhost:8080/"
1397         },
1398         "id": "rk01vrHU4vLo",
1399         "outputId": "5193081f-f0d5-417a-9b6e-6d3951706315"
1400     },

```

```
1401     "execution_count": 38,
1402     "outputs": [
1403     {
1404         "output_type": "execute_result",
1405         "data": {
1406             "text/plain": [
1407                 "0    1\n",
1408                 "1    0\n",
1409                 "2    1\n",
1410                 "3    0\n",
1411                 "4    0\n",
1412                 "Name: Exited, dtype: int64"
1413             ]
1414         },
1415         "metadata": {},
1416         "execution_count": 38
1417     }
1418 ]
1419 },
1420 {
1421     "cell_type": "code",
1422     "source": [
1423         "from sklearn.preprocessing import StandardScaler\n",
1424         "scaler=StandardScaler()\n",
1425         "x=scaler.fit_transform(x)"
1426     ],
1427     "metadata": {
1428         "id": "3aVuJwYg4xnR"
1429     },
1430     "execution_count": 39,
1431     "outputs": []
1432 },
1433 {
1434     "cell_type": "code",
1435     "source": [
1436         "x"
```



```

1437 ],
1438 "metadata": {
1439   "colab": {
1440     "base_uri": "https://localhost:8080/"
1441   },
1442   "id": "FUABWKY6400a",
1443   "outputId": "1830203b-397b-49b9-d5a2-f996f61debef"
1444 },
1445 "execution_count": 40,
1446 "outputs": [
1447   {
1448     "output_type": "execute_result",
1449     "data": {
1450       "text/plain": [
1451         "array([[ -0.32687761, -0.90188624, -1.09598752, ...,  0.64609167,\n",
1452         "         "      0.97024255,  0.02188649],\n",
1453         "         [ -0.44080365,  1.51506738, -1.09598752, ..., -1.54776799,\n",
1454         "         "      0.97024255,  0.21653375],\n",
1455         "         [ -1.53863634, -0.90188624, -1.09598752, ...,  0.64609167,\n",
1456         "         "      -1.03067011,  0.2406869 ],\n",
1457         "         ..., \n",
1458         "         [  0.60524449, -0.90188624, -1.09598752, ..., -1.54776799,\n",
1459         "         "      0.97024255, -1.00864308],\n",
1460         "         [  1.25772996,  0.30659057,  0.91241915, ...,  0.64609167,\n",
1461         "         "      -1.03067011, -0.12523071],\n",
1462         "         [  1.4648682 , -0.90188624, -1.09598752, ...,  0.64609167,\n",
1463         "         "      -1.03067011, -1.07636976]])"
1464       ]
1465     },
1466     "metadata": {},
1467     "execution_count": 40
1468   }
1469 ]
1470 },
1471 {
1472   "cell_type": "code",

```

```
1473 "source": [  
1474     "from sklearn.model_selection import train_test_split\n",  
1475     "x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33)\n",  
1476     "x_train.shape"  
1477 ],  
1478 "metadata": {  
1479     "colab": {  
1480         "base_uri": "https://localhost:8080/"  
1481     },  
1482     "id": "W5RG9ibx43hT",  
1483     "outputId": "f7fea99c-04f2-4b96-eac8-586082d0c5be"  
1484 },  
1485 "execution_count": 41,  
1486 "outputs": [  
1487     {  
1488         "output_type": "execute_result",  
1489         "data": {  
1490             "text/plain": [  
1491                 "(6700, 10)"  
1492             ]  
1493         },  
1494         "metadata": {},  
1495         "execution_count": 41  
1496     }  
1497 ],  
1498 },  
1499 {  
1500     "cell_type": "code",  
1501     "source": [  
1502         "x_test.shape"  
1503     ],  
1504     "metadata": {  
1505         "colab": {  
1506             "base_uri": "https://localhost:8080/"  
1507         },  
1508         "id": "eVLr7zta47gq",
```

```
1509     "outputId": "1b65fe69-ad5f-4af0-e7bb-2a91aa5374ea"
1510 },
1511 "execution_count": 42,
1512 "outputs": [
1513   {
1514     "output_type": "execute_result",
1515     "data": {
1516       "text/plain": [
1517         "(3300, 10)"
1518       ]
1519     },
1520     "metadata": {},
1521     "execution_count": 42
1522   }
1523 ],
1524 },
1525 {
1526   "cell_type": "code",
1527   "source": [
1528     "y_train.shape"
1529   ],
1530   "metadata": {
1531     "colab": {
1532       "base_uri": "https://localhost:8080/"
1533     },
1534     "id": "JqH9-c4d471-",
1535     "outputId": "05d84cb3-1157-4821-c958-9fa0a3d0ad28"
1536   },
1537   "execution_count": 43,
1538   "outputs": [
1539     {
1540       "output_type": "execute_result",
1541       "data": {
1542         "text/plain": [
1543           "(6700,)"
1544         ]
1545       }
1546     }
1547   ]
1548 }
```

```
1545         },
1546         "metadata": {},
1547         "execution_count": 43
1548     }
1549 ]
1550 },
1551 {
1552     "cell_type": "code",
1553     "source": [
1554         "y_test.shape"
1555     ],
1556     "metadata": {
1557         "colab": {
1558             "base_uri": "https://localhost:8080/"
1559         },
1560         "id": "mEZHXROH4_lg",
1561         "outputId": "3cd8ec97-8ad3-49b0-ae1-3b42e8b370eb"
1562     },
1563     "execution_count": 44,
1564     "outputs": [
1565         {
1566             "output_type": "execute_result",
1567             "data": {
1568                 "text/plain": [
1569                     "(3300,)"
1570                 ]
1571             },
1572             "metadata": {},
1573             "execution_count": 44
1574         }
1575     ]
1576 }
```

```
1577 ],
1578 "metadata": {
1579     "accelerator": "GPU",
1580     "colab": {
1581         "collapsed_sections": [],
1582         "provenance": []
1583     },
1584     "kernel_spec": {
1585         "display_name": "Python 3",
1586         "name": "python3"
1587     }
1588 },
1589 "nbformat": 4,
1590 "nbformat_minor": 0
1591 }
```