

ASSIGNMENT -2
Python
Programming

Team ID	PNT2022TMID14463
Project Name	Real Time Communication System Powered By AI For Specially Abled
Roll No	711319EC107

Question-1 :

1 . Importing Required Package

Solution :

```
import pandas as pd
import seaborn as sns
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
```

Question-2 :

2. Loading the Dataset

Solution :

```
df = pd.read_csv("/content/Churn_Modelling.csv")
```

```
df
```

Output:



	RowNumber	CustomerID	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	1	15634602	Hargrave	615	France	Female	42	2	0.00	1	1	1	101345.88	1
1	2	10647311	Hill	806	Spain	Female	41	1	83007.86	1	0	1	112542.56	0
2	3	15618304	Onio	302	France	Female	42	8	159650.80	3	1	0	113431.57	1
3	4	15701354	Boni	599	France	Female	39	1	0.00	2	0	0	93826.63	0
4	5	15707956	Mitchell	950	Spain	Female	43	2	125515.82	1	1	1	79004.10	0
...
9995	9996	15606229	Ogibaku	771	France	Male	38	5	0.00	2	1	0	96270.64	0
9996	9997	15669882	Jonnstone	516	France	Male	35	10	57369.61	1	1	1	101694.77	0
9997	9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42093.58	1
9998	9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	82898.52	1
9999	10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

10000 rows x 14 columns

3. Visualizations

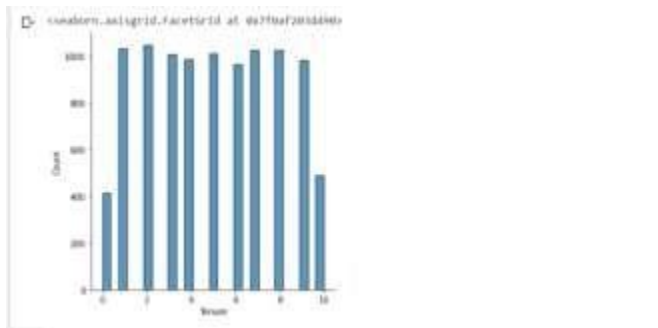
Question-3 :

3.1 Univariate Analysis

Solution:

```
sns.displot(df.Tenure)
```

Output:

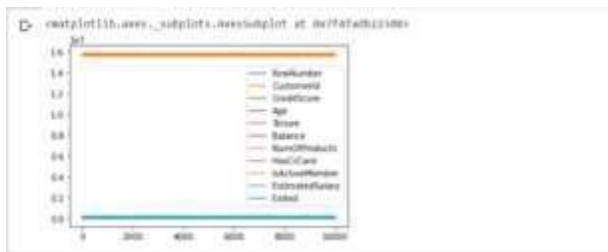


3.2 Bi-Variate Analysis

Solution:

```
df.plot.line()
```

Output:



3.3 Multi - Variate Analysis

Solution:

```
sns.lmplot("Age", "NumOfProducts", df, hue="NumOfProducts", fit_reg=False);
```

Output:



4. Perform descriptive statistics on the dataset.

Question-4 :

Solution:

```
df.describe()
```

Output:

	RouteNumber	CustomerID	CreditScore	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
count	10000.00000	1.000000e+04	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000	10000.00000
mean	3000.50000	1.500000e+07	480.52800	38.52180	0.01200	76483.66098	1.53200	0.70000	0.51510	100000.23488	0.20370
std	2000.89500	7.183618e+04	98.65329	10.40700	2.892174	62397.40292	0.58184	0.45994	0.488797	57510.40258	0.402708
min	1.00000	1.556179e+07	355.00000	18.00000	0.00000	0.00000	1.00000	0.00000	0.00000	11.58000	0.00000
25%	2500.75000	1.562853e+07	384.00000	32.00000	3.00000	0.00000	1.00000	0.00000	0.00000	51002.10000	0.00000
50%	3000.50000	1.568074e+07	452.00000	37.00000	5.00000	87188.54000	1.00000	1.00000	1.00000	100101.81500	0.00000
75%	3500.25000	1.575323e+07	518.00000	44.00000	7.00000	127044.24000	2.00000	1.00000	1.00000	146188.24750	0.00000
max	10000.00000	1.587591e+07	850.00000	62.00000	10.00000	250000.00000	4.00000	1.00000	1.00000	199902.48000	1.00000

5. Handle the Missing values.

Question-5 :

Solution:

```
data = pd.read_csv("Churn_Modelling.csv")
pd.isnull(data["Gender"])
```

Output:



The output shows a series of 10000 boolean values, all of which are False, indicating that there are no missing values in the 'Gender' column. The output is truncated with '...' in the middle. The final line shows the name of the series as 'Gender', its length as 10000, and its dtype as bool.

Question-6:

6. Find the outliers and replace the outliers.

Solution:

```
df["Tenure"] = np.where(df["Tenure"] > 10, np.median(df["Tenure"]),
df["Tenure"])
```

Output:



The output shows the 'Tenure' column after replacing outliers. The values are mostly 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000. The output is truncated with '...' in the middle. The final line shows the name of the series as 'Tenure', its length as 10000, and its dtype as object.

Question-7 :

7. Check for Categorical columns and perform encoding.

Solution:

```
pd.get_dummies(df, columns=["Gender", "Age"], prefix=["Age", "Gender"])
).head()
```

Output:

	RowNumber	CustomerId	Surname	CreditScore	Geography	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	...	Gender_78
0	1	15634002	Hargrave	619	France	2	0.00	1	1	1	...	0
1	2	15647311	Hill	608	Spain	1	83607.86	1	0	1	...	0
2	3	15619304	Onio	502	France	8	159680.80	3	1	0	...	0
3	4	15701354	Boni	699	France	1	0.00	2	0	0	...	0
4	5	15737800	Mitchell	650	Spain	2	125510.62	1	1	1	...	0

5 rows × 14 columns

Output:

	HasCrCard	IsActiveMember	...	Gender_78	Gender_79	Gender_80	Gender_81	Gender_82	Gender_83	Gender_84	Gender_85	Gender_86	Gender_92
	1	1	...	0	0	0	0	0	0	0	0	0	0
	0	1	...	0	0	0	0	0	0	0	0	0	0
	1	0	...	0	0	0	0	0	0	0	0	0	0
	0	0	...	0	0	0	0	0	0	0	0	0	0
	1	1	...	0	0	0	0	0	0	0	0	0	0

5 rows × 14 columns

Question-8:

1. Split the data into dependent and independent variables

1.Split the data into Independent variables.

Solution:

```
X = df.iloc[:, :-2].values
print(X)
```

Output:

```
[[1 15634682 'Hargrave' ... 1 1 1]
 [2 15647311 'Hill' ... 1 0 1]
 [3 15619384 'Orio' ... 1 1 0]
 ...
 [9998 15584532 'Liu' ... 1 0 1]
 [9999 15481355 'Sabbatini' ... 2 1 0]
 [10000 15628319 'Walker' ... 1 1 0]]
```

8.2 Split the data into Dependent variables.

Solution:

```
Y = df.iloc[:, -1].values
print(Y)
```

Output:

```
[1 0 1 ... 1 1 0]
```

Question-9 :

9. Scale the independent variables

Solution:

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df[["RowNumber"]] = scaler.fit_transform(df[["RowNumber"]])
print(df)
```

Output:

	Author	Lastname	Country	Continent	Geography	Gender	Age
0	0.0000	17030002	Spain	Europe	Spain	Female	41
1	0.0000	15657311	Italy	Europe	Spain	Female	41
2	0.0000	15629000	Spain	Europe	Spain	Female	42
3	0.0000	17051204	Spain	Europe	Spain	Female	43
4	0.0000	17170000	Spain	Europe	Spain	Female	43
5	0.0000	15000220	Spain	Europe	Spain	Female	44
6	0.0000	15559602	Spain	Europe	Spain	Female	45
7	0.0000	17000452	Spain	Europe	Spain	Female	46
8	0.0000	15000210	Spain	Europe	Spain	Female	47
9	0.0000	15620310	Spain	Europe	Spain	Female	48
10	0.0000	15000220	Spain	Europe	Spain	Female	49
11	0.0000	15000220	Spain	Europe	Spain	Female	50
12	0.0000	15000220	Spain	Europe	Spain	Female	51
13	0.0000	15000220	Spain	Europe	Spain	Female	52
14	0.0000	15000220	Spain	Europe	Spain	Female	53
15	0.0000	15000220	Spain	Europe	Spain	Female	54
16	0.0000	15000220	Spain	Europe	Spain	Female	55
17	0.0000	15000220	Spain	Europe	Spain	Female	56
18	0.0000	15000220	Spain	Europe	Spain	Female	57
19	0.0000	15000220	Spain	Europe	Spain	Female	58
20	0.0000	15000220	Spain	Europe	Spain	Female	59
21	0.0000	15000220	Spain	Europe	Spain	Female	60
22	0.0000	15000220	Spain	Europe	Spain	Female	61
23	0.0000	15000220	Spain	Europe	Spain	Female	62
24	0.0000	15000220	Spain	Europe	Spain	Female	63
25	0.0000	15000220	Spain	Europe	Spain	Female	64
26	0.0000	15000220	Spain	Europe	Spain	Female	65
27	0.0000	15000220	Spain	Europe	Spain	Female	66
28	0.0000	15000220	Spain	Europe	Spain	Female	67
29	0.0000	15000220	Spain	Europe	Spain	Female	68
30	0.0000	15000220	Spain	Europe	Spain	Female	69
31	0.0000	15000220	Spain	Europe	Spain	Female	70
32	0.0000	15000220	Spain	Europe	Spain	Female	71
33	0.0000	15000220	Spain	Europe	Spain	Female	72
34	0.0000	15000220	Spain	Europe	Spain	Female	73
35	0.0000	15000220	Spain	Europe	Spain	Female	74
36	0.0000	15000220	Spain	Europe	Spain	Female	75
37	0.0000	15000220	Spain	Europe	Spain	Female	76
38	0.0000	15000220	Spain	Europe	Spain	Female	77
39	0.0000	15000220	Spain	Europe	Spain	Female	78
40	0.0000	15000220	Spain	Europe	Spain	Female	79
41	0.0000	15000220	Spain	Europe	Spain	Female	80
42	0.0000	15000220	Spain	Europe	Spain	Female	81
43	0.0000	15000220	Spain	Europe	Spain	Female	82
44	0.0000	15000220	Spain	Europe	Spain	Female	83
45	0.0000	15000220	Spain	Europe	Spain	Female	84
46	0.0000	15000220	Spain	Europe	Spain	Female	85
47	0.0000	15000220	Spain	Europe	Spain	Female	86
48	0.0000	15000220	Spain	Europe	Spain	Female	87
49	0.0000	15000220	Spain	Europe	Spain	Female	88
50	0.0000	15000220	Spain	Europe	Spain	Female	89
51	0.0000	15000220	Spain	Europe	Spain	Female	90
52	0.0000	15000220	Spain	Europe	Spain	Female	91
53	0.0000	15000220	Spain	Europe	Spain	Female	92
54	0.0000	15000220	Spain	Europe	Spain	Female	93
55	0.0000	15000220	Spain	Europe	Spain	Female	94
56	0.0000	15000220	Spain	Europe	Spain	Female	95
57	0.0000	15000220	Spain	Europe	Spain	Female	96
58	0.0000	15000220	Spain	Europe	Spain	Female	97
59	0.0000	15000220	Spain	Europe	Spain	Female	98
60	0.0000	15000220	Spain	Europe	Spain	Female	99
61	0.0000	15000220	Spain	Europe	Spain	Female	100
62	0.0000	15000220	Spain	Europe	Spain	Female	101
63	0.0000	15000220	Spain	Europe	Spain	Female	102
64	0.0000	15000220	Spain	Europe	Spain	Female	103
65	0.0000	15000220	Spain	Europe	Spain	Female	104
66	0.0000	15000220	Spain	Europe	Spain	Female	105
67	0.0000	15000220	Spain	Europe	Spain	Female	106
68	0.0000	15000220	Spain	Europe	Spain	Female	107
69	0.0000	15000220	Spain	Europe	Spain	Female	108
70	0.0000	15000220	Spain	Europe	Spain	Female	109
71	0.0000	15000220	Spain	Europe	Spain	Female	110
72	0.0000	15000220	Spain	Europe	Spain	Female	111
73	0.0000	15000220	Spain	Europe	Spain	Female	112
74	0.0000	15000220	Spain	Europe	Spain	Female	113
75	0.0000	15000220	Spain	Europe	Spain	Female	114
76	0.0000	15000220	Spain	Europe	Spain	Female	115
77	0.0000	15000220	Spain	Europe	Spain	Female	116
78	0.0000	15000220	Spain	Europe	Spain	Female	117
79	0.0000	15000220	Spain	Europe	Spain	Female	118
80	0.0000	15000220	Spain	Europe	Spain	Female	119
81	0.0000	15000220	Spain	Europe	Spain	Female	120
82	0.0000	15000220	Spain	Europe	Spain	Female	121
83	0.0000	15000220	Spain	Europe	Spain	Female	122
84	0.0000	15000220	Spain	Europe	Spain	Female	123
85	0.0000	15000220	Spain	Europe	Spain	Female	124
86	0.0000	15000220	Spain	Europe	Spain	Female	125
87	0.0000	15000220	Spain	Europe	Spain	Female	126
88	0.0000	15000220	Spain	Europe	Spain	Female	127
89	0.0000	15000220	Spain	Europe	Spain	Female	128
90	0.0000	15000220	Spain	Europe	Spain	Female	129
91	0.0000	15000220	Spain	Europe	Spain	Female	130
92	0.0000	15000220	Spain	Europe	Spain	Female	131
93	0.0000	15000220	Spain	Europe	Spain	Female	132
94	0.0000	15000220	Spain	Europe	Spain	Female	133
95	0.0000	15000220	Spain	Europe	Spain	Female	134
96	0.0000	15000220	Spain	Europe	Spain	Female	135
97	0.0000	15000220	Spain	Europe	Spain	Female	136
98	0.0000	15000220	Spain	Europe	Spain	Female	137
99	0.0000	15000220	Spain	Europe	Spain	Female	138
100	0.0000	15000220	Spain	Europe	Spain	Female	139
101	0.0000	15000220	Spain	Europe	Spain	Female	140
102	0.0000	15000220	Spain	Europe	Spain	Female	141
103	0.0000	15000220	Spain	Europe	Spain	Female	142
104	0.0000	15000220	Spain	Europe	Spain	Female	143
105	0.0000	15000220	Spain	Europe	Spain	Female	144
106	0.0000	15000220	Spain	Europe	Spain	Female	145
107	0.0000	15000220	Spain	Europe	Spain	Female	146
108	0.0000	15000220	Spain	Europe	Spain	Female	147
109	0.0000	15000220	Spain	Europe	Spain	Female	148
110	0.0000	15000220	Spain	Europe	Spain	Female	149
111	0.0000	15000220	Spain	Europe	Spain	Female	150
112	0.0000	15000220	Spain	Europe	Spain	Female	151
113	0.0000	15000220	Spain	Europe	Spain	Female	152
114	0.0000	15000220	Spain	Europe	Spain	Female	153
115	0.0000	15000220	Spain	Europe	Spain	Female	154
116	0.0000	15000220	Spain	Europe	Spain	Female	155
117	0.0000	15000220	Spain	Europe	Spain	Female	156
118	0.0000	15000220	Spain	Europe	Spain	Female	157
119	0.0000	15000220	Spain	Europe	Spain	Female	158
120	0.0000	15000220	Spain	Europe	Spain	Female	159
121	0.0000	15000220	Spain	Europe	Spain	Female	160
122	0.0000	15000220	Spain	Europe	Spain	Female	161
123	0.0000	15000220	Spain	Europe	Spain	Female	162
124	0.0000	15000220	Spain	Europe	Spain	Female	163
125	0.0000	15000220	Spain	Europe	Spain	Female	164
126	0.0000	15000220	Spain	Europe	Spain	Female	165
127	0.0000	15000220	Spain	Europe	Spain	Female	166
128	0.0000	15000220	Spain	Europe	Spain	Female	167
129	0.0000	15000220	Spain	Europe	Spain	Female	168
130	0.0000	15000220	Spain	Europe	Spain	Female	169
131	0.0000	15000220	Spain	Europe	Spain	Female	170
132	0.0000	15000220	Spain	Europe	Spain	Female	171
133	0.0000	15000220	Spain	Europe	Spain	Female	172
134	0.0000	15000220	Spain	Europe	Spain	Female	173
135	0.0000	15000220	Spain	Europe	Spain	Female	174
136	0.0000	15000220	Spain	Europe	Spain	Female	175
137	0.0000	15000220	Spain	Europe	Spain	Female	176
138	0.0000	15000220	Spain	Europe	Spain	Female	177
139	0.0000	15000220	Spain	Europe	Spain	Female	178
140	0.0000	15000220	Spain	Europe	Spain	Female	179
141	0.0000	15000220	Spain	Europe	Spain	Female	180
142	0.0000	15000220	Spain	Europe	Spain	Female	181
143	0.0000	15000220	Spain	Europe	Spain	Female	182
144	0.0000	15000220	Spain	Europe	Spain	Female	183
145	0.0000	15000220	Spain	Europe	Spain	Female	184
146	0.0000	15000220	Spain	Europe	Spain	Female	185
147	0.0000	15000220	Spain	Europe	Spain	Female	186
148	0.0000	15000220	Spain	Europe	Spain	Female	187
149	0.0000	15000220	Spain	Europe	Spain	Female	188
150	0.0000	15000220	Spain	Europe	Spain	Female	189
151	0.0000	15000220	Spain	Europe	Spain	Female	190
152	0.0000	15000220	Spain	Europe	Spain	Female	191
153	0.0000	15000220	Spain	Europe	Spain	Female	192
154	0.0000	15000220	Spain	Europe	Spain	Female	193
155	0.0000	15000220	Spain	Europe	Spain	Female	194
156	0.0000	15000220	Spain	Europe	Spain	Female	195
157	0.0000	15000220	Spain	Europe	Spain	Female	196
158	0.0000	15000220	Spain	Europe	Spain	Female	197
159	0.0000	15000220	Spain	Europe	Spain	Female	198
160	0.0000	15000220	Spain	Europe	Spain	Female	199
161	0.0000	15000220	Spain	Europe	Spain	Female	200
162	0.0000	15000220	Spain	Europe	Spain	Female	201
163	0.0000	15000220	Spain	Europe	Spain	Female	202
164	0.0000	15000220	Spain	Europe	Spain	Female	203
165	0.0000	15000220	Spain	Europe	Spain	Female	204
166	0.0000	15000220	Spain	Europe	Spain	Female	205
167	0.0000	15000220	Spain	Europe	Spain	Female	206
168	0.0000	15000220	Spain	Europe	Spain	Female	207
169	0.0000	15000220	Spain	Europe	Spain	Female	208
170	0.0000	15000220	Spain	Europe	Spain	Female	209
171	0.0000	15000220	Spain	Europe	Spain	Female	210
172	0.0000	15000220	Spain	Europe	Spain	Female	211
173	0.0000	15000220	Spain	Europe	Spain	Female	212
174	0.0000	15000220	Spain	Europe	Spain	Female	213
175	0.0000	15000220	Spain	Europe	Spain	Female	214
176	0.0000	15000220	Spain	Europe	Spain	Female	215
177	0.0000	15000220	Spain	Europe	Spain	Female	216
178	0.0000	15000220	Spain	Europe	Spain	Female	217
179	0.0000	15000220	Spain	Europe	Spain	Female	218
180	0.0000	15000220	Spain	Europe	Spain	Female	219
181	0.0000	15000220	Spain	Europe	Spain	Female	220
182	0.0000	15000220	Spain	Europe	Spain	Female	221
183	0.0000	15000220	Spain	Europe	Spain	Female	

Question-10 :

10. Split the data into training and testing

Solution:

```
from sklearn.model_selection import train_test_split
train_size=0.8
X = df.drop(columns = ['Tenure']).copy()
y = df['Tenure']
X_train, X_rem, y_train, y_rem = train_test_split(X,y, train_size=0.8)
test_size = 0.5
X_valid, X_test, y_valid, y_test = train_test_split(X_rem,y_rem, test_size=0.5)
print(X_train.shape), print(y_train.shape)
print(X_valid.shape), print(y_valid.shape)
print(X_test.shape), print(y_test.shape)
```

Output:

```
D: (8000, 13)
(8000, )
(1000, 13)
(1000, )
(1000, 13)
(1000, )
(None, None)
```


TEAM LEADER: S.THARANI

TEAM MEMBERS:

1. K.V.SUNSHETHA

2. L.P.SINDHUJA

3. S.SRINIVASAN

TEAM ID :PNT2022TMID14463