

```
In [ ]: 1 import tensorflow as tf
2 from tensorflow import keras
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6 import seaborn as sns
7 import math
8 from sklearn.preprocessing import LabelEncoder
9 from sklearn.preprocessing import MinMaxScaler
10 from sklearn.model_selection import train_test_split
```

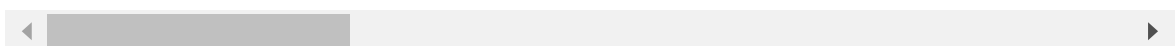
```
In [ ]: 1 data = pd.read_csv("/content/drive/MyDrive/watson_healthcare_modified.csv")
```

```
In [ ]: 1 data
```

```
Out[43]:
```

	EmployeeID	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHom
0	1313919	41	No	Travel_Rarely	1102	Cardiology	
1	1200302	49	No	Travel_Frequently	279	Maternity	
2	1060315	37	Yes	Travel_Rarely	1373	Maternity	
3	1272912	33	No	Travel_Frequently	1392	Maternity	
4	1414939	27	No	Travel_Rarely	591	Maternity	
...	...	...	...	...	...	...	..
1671	1117656	26	Yes	Travel_Rarely	471	Neurology	2
1672	1152327	46	No	Travel_Rarely	1125	Cardiology	1
1673	1812428	20	No	Travel_Rarely	959	Maternity	
1674	1812429	39	No	Travel_Rarely	466	Neurology	
1675	1152329	27	No	Travel_Rarely	511	Cardiology	

1676 rows × 35 columns



In []:

1 data.isnull().sum()

Out[44]:

EmployeeID	0
Age	0
Attrition	0
BusinessTravel	0
DailyRate	0
Department	0
DistanceFromHome	0
Education	0
EducationField	0
EmployeeCount	0
EnvironmentSatisfaction	0
Gender	0
HourlyRate	0
JobInvolvement	0
JobLevel	0
JobRole	0
JobSatisfaction	0
MaritalStatus	0
MonthlyIncome	0
MonthlyRate	0
NumCompaniesWorked	0
Over18	0
OverTime	0
PercentSalaryHike	0
PerformanceRating	0
RelationshipSatisfaction	0
StandardHours	0
Shift	0
TotalWorkingYears	0
TrainingTimesLastYear	0
WorkLifeBalance	0
YearsAtCompany	0
YearsInCurrentRole	0
YearsSinceLastPromotion	0
YearsWithCurrManager	0

dtype: int64

```
In [ ]: 1 data.dtypes
```

```
Out[45]: EmployeeID      int64
Age      int64
Attrition object
BusinessTravel object
DailyRate int64
Department object
DistanceFromHome int64
Education int64
EducationField object
EmployeeCount int64
EnvironmentSatisfaction int64
Gender object
HourlyRate int64
JobInvolvement int64
JobLevel int64
JobRole object
JobSatisfaction int64
MaritalStatus object
MonthlyIncome int64
MonthlyRate int64
NumCompaniesWorked int64
Over18 object
OverTime object
PercentSalaryHike int64
PerformanceRating int64
RelationshipSatisfaction int64
StandardHours int64
Shift int64
TotalWorkingYears int64
TrainingTimesLastYear int64
WorkLifeBalance int64
YearsAtCompany int64
YearsInCurrentRole int64
YearsSinceLastPromotion int64
YearsWithCurrManager int64
dtype: object
```

```
In [ ]: 1 data.drop("EmployeeID", axis =1, inplace=True)
```

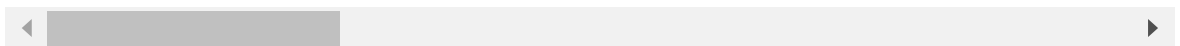
In []:

```
1 data
```

Out[47]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education
0	41	No	Travel_Rarely	1102	Cardiology	1	2
1	49	No	Travel_Frequently	279	Maternity	8	1
2	37	Yes	Travel_Rarely	1373	Maternity	2	2
3	33	No	Travel_Frequently	1392	Maternity	3	4
4	27	No	Travel_Rarely	591	Maternity	2	1
...	...	...	...	...	...	...	...
1671	26	Yes	Travel_Rarely	471	Neurology	24	3
1672	46	No	Travel_Rarely	1125	Cardiology	10	3
1673	20	No	Travel_Rarely	959	Maternity	1	3
1674	39	No	Travel_Rarely	466	Neurology	1	1
1675	27	No	Travel_Rarely	511	Cardiology	2	2

1676 rows × 34 columns



In []:

```
1 data.Over18.values
```

Out[48]: array(['Y', 'Y', 'Y', ..., 'Y', 'Y', 'Y'], dtype=object)

In []:

```
1 data.OverTime.values
```

Out[49]: array(['Yes', 'No', 'Yes', ..., 'No', 'No', 'No'], dtype=object)

In []:

```
1 def col_unique_value(d):  
2     for col in d:  
3         print(col,":", d[col].unique())
```

In []:

```
1 col_unique_value(data)
```

Age : [41 49 37 33 27 32 59 30 38 36 35 29 31 34 28 22 53 24 21 42 44 46 39 43  
50 26 48 55 45 56 23 51 40 54 58 20 25 19 57 52 47 18 60]  
Attrition : ['No' 'Yes']  
BusinessTravel : ['Travel\_Rarely' 'Travel\_Frequently' 'Non-Travel']  
DailyRate : [1102 279 1373 1392 591 1005 1324 1358 216 1299 809 153 670 1346  
103 1389 334 1123 1219 371 673 1218 419 391 699 1282 1125 691  
477 705 924 1459 125 895 813 1273 869 890 852 1141 464 1240  
1357 994 721 1360 1065 408 1211 1229 626 1434 1488 1097 1443 515  
853 1142 655 1115 427 653 989 1435 1223 836 1195 1339 664 318  
1225 1328 1082 548 132 746 776 193 397 945 1214 111 573 1153  
1400 541 432 288 669 530 632 1334 638 1093 1217 1353 120 682  
489 807 827 871 665 1040 1420 240 1280 534 1456 658 142 1127  
1031 1189 1354 1467 922 394 1312 750 441 684 249 841 147 528  
594 470 957 542 802 1355 1150 1329 959 1033 1316 364 438 689  
201 1427 857 933 1181 1395 662 1436 194 967 1496 1169 1145 630  
303 1256 440 1450 1452 465 702 1157 602 1480 1268 713 134 526  
1380 140 629 1356 328 1084 931 692 1069 313 894 556 1344 290  
138 926 1261 472 1002 878 905 1180 121 1136 635 1151 644 1045  
829 1242 1469 896 992 1052 1147 1396 663 119 979 319 1413 944  
1323 532 818 854 1034 771 1401 1431 976 1411 1300 252 1327 832  
1017 1199 504 505 916 1247 685 269 1416 833 307 1311 128 488  
529 1210 1463 675 1385 1403 452 666 1158 228 996 728 1315 322  
1479 797 1070 442 496 1372 920 688 1449 1117 636 506 444 950  
889 555 230 1232 566 1302 812 1476 218 1132 1105 906 849 390  
106 1249 192 553 117 185 1091 723 1220 588 1377 1018 1275 798  
672 1162 508 1482 559 210 928 1001 549 1124 738 570 1130 1192  
343 144 1296 1309 483 810 544 1062 1319 641 1332 756 845 593  
1171 350 921 1144 143 1046 575 156 1283 755 304 1178 329 1362  
1371 202 253 164 1107 759 1305 982 821 1381 480 1473 891 1063  
645 1490 317 422 1485 1368 1448 296 1398 1349 986 1099 1116 1499  
983 1009 1303 1274 1277 587 413 1276 988 1474 163 267 619 302  
443 828 561 426 232 1306 1094 509 775 195 258 471 799 956  
535 1495 446 1245 703 823 1246 622 1287 448 254 1365 538 525  
558 782 362 1236 1112 204 1343 604 1216 646 160 238 1397 306  
991 482 1176 913 1076 727 885 243 806 817 1410 1207 1442 693  
929 562 608 580 970 1179 294 314 316 654 168 381 217 501  
650 141 804 975 1090 346 430 268 167 621 527 883 954 310  
719 725 715 657 1146 182 376 571 384 791 1111 1243 1092 1325  
805 213 118 676 1252 286 1258 932 1041 859 720 946 1184 436  
589 760 887 1318 625 180 586 1012 661 930 342 1230 1271 1278  
607 130 300 583 1418 1269 379 395 1265 1222 341 868 1231 102  
881 1383 1075 374 1086 781 177 500 1425 1454 617 1085 995 1122  
618 546 462 1198 1272 154 1137 1188 188 1333 867 263 938 129  
616 498 1404 1053 289 1376 231 152 882 903 1379 335 722 461  
974 1126 840 1134 248 955 939 1391 1206 287 1441 109 1066 277  
1055 265 135 247 1035 266 145 1038 1234 1109 1089 788 124 660  
1186 1464 796 415 769 1003 1366 330 1492 1204 309 1330 469 697  
1262 1050 770 406 203 1308 984 439 793 1451 1182 174 490 718  
433 773 603 874 367 199 481 647 1384 902 819 862 1457 977  
942 1402 1421 1361 917 200 150 179 696 116 363 107 1465 458  
1212 1103 966 1010 326 1098 969 1167 694 1320 536 373 599 251  
131 237 1429 648 735 531 429 968 879 640 412 848 360 1138  
325 1322 299 1030 634 524 256 1060 935 495 282 206 943 523  
507 601 855 1291 1405 1369 999 1202 285 404 736 1498 1200 1439  
499 205 683 1462 949 652 332 1475 337 971 1174 667 560 172  
383 1255 359 401 377 592 1445 1221 866 981 447 1326 748 990  
405 115 790 830 1193 1423 467 271 410 1083 516 224 136 1029  
333 1440 674 1342 898 824 492 598 740 888 1288 104 1108 479  
1351 474 437 884 1370 264 1059 563 457 1313 241 1015 336 1387  
170 208 671 711 737 1470 365 763 567 486 772 301 311 584  
880 392 148 708 1259 786 370 678 146 581 918 1238 585 741

552 369 717 543 964 792 611 176 897 600 1054 428 181 211  
 1079 590 305 953 478 1375 244 511 1294 196 734 1239 1253 1128  
 1336 234 766 261 1194 431 572 1422 1297 574 355 207 706 280  
 726 414 352 1224 459 1254 1131 835 1172 1266 783 219 1213 1096  
 1251 1394 605 1064 1337 937 157 754 1168 155 1444 189 911 1321  
 1154 557 642 801 161 1382 1037 105 582 704 345 1120 1378 468  
 613 1023 628 466]  
 Department : ['Cardiology' 'Maternity' 'Neurology']  
 DistanceFromHome : [ 1 8 2 3 24 23 27 16 15 26 19 21 5 11 9 7 6 10 4 25 12 18 29 22  
 14 20 28 17 13]  
 Education : [2 1 4 3 5]  
 EducationField : ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'  
 'Human Resources']  
 EmployeeCount : [1]  
 EnvironmentSatisfaction : [2 3 4 1]  
 Gender : ['Female' 'Male']  
 HourlyRate : [ 94 61 92 56 40 79 81 67 44 84 49 31 93 50 51 80 96 78  
 45 82 53 83 58 72 48 42 41 86 97 75 33 37 73 98 36 47  
 71 30 43 99 59 95 57 76 87 66 55 32 52 70 62 64 63 60  
 100 46 39 77 35 91 54 34 90 65 88 85 89 68 69 74 38]  
 JobInvolvement : [3 2 4 1]  
 JobLevel : [2 1 3 4 5]  
 JobRole : ['Nurse' 'Other' 'Therapist' 'Administrative' 'Admin']  
 JobSatisfaction : [4 2 3 1]  
 MaritalStatus : ['Single' 'Married' 'Divorced']  
 MonthlyIncome : [ 5993 5130 2090 ... 12742 9071 2836]  
 MonthlyRate : [19479 24907 2396 ... 11563 11757 26997]  
 NumCompaniesWorked : [8 1 6 9 0 4 5 2 7 3]  
 Over18 : ['Y']  
 OverTime : ['Yes' 'No']  
 PercentSalaryHike : [11 23 15 12 13 20 22 21 17 14 16 18 19 24 25]  
 PerformanceRating : [3 4]  
 RelationshipSatisfaction : [1 4 2 3]  
 StandardHours : [80]  
 Shift : [0 1 3 2]  
 TotalWorkingYears : [ 8 10 7 6 12 1 17 5 3 31 13 0 26 24 22 9 19 2 23 14 15 4 29 28  
 21 25 20 11 16 37 38 30 40 18 36 34 32 33 35 27]  
 TrainingTimesLastYear : [0 3 2 5 1 4 6]  
 WorkLifeBalance : [1 3 2 4]  
 YearsAtCompany : [ 6 10 0 8 2 7 1 9 5 4 25 3 12 14 22 15 27 21 17 11 13 37 16 20  
 40 24 33 19 36 18 29 31 32 34 26 30 23]  
 YearsInCurrentRole : [ 4 7 0 2 5 9 8 3 6 13 1 15 14 16 11 10 12 18 17]  
 YearsSinceLastPromotion : [ 0 1 3 2 7 4 8 6 5 15 9 13 12 10 11 14]  
 YearsWithCurrManager : [ 5 7 0 2 6 8 3 11 17 1 4 12 9 10 15 13 16 14]

```

In [:]
1 yes_no_col = ['OverTime','Attrition']
2 for i in yes_no_col:
3     data[i].replace({"Yes":1,"No":0},inplace=True)
  
```

In []:

```
1 col_unique_value(data)
```



Age : [41 49 37 33 27 32 59 30 38 36 35 29 31 34 28 22 53 24 21 42 44 46 39 43  
50 26 48 55 45 56 23 51 40 54 58 20 25 19 57 52 47 18 60]  
Attrition : [0 1]  
BusinessTravel : ['Travel\_Rarely' 'Travel\_Frequently' 'Non-Travel']  
DailyRate : [1102 279 1373 1392 591 1005 1324 1358 216 1299 809 153 670 1346  
103 1389 334 1123 1219 371 673 1218 419 391 699 1282 1125 691  
477 705 924 1459 125 895 813 1273 869 890 852 1141 464 1240  
1357 994 721 1360 1065 408 1211 1229 626 1434 1488 1097 1443 515  
853 1142 655 1115 427 653 989 1435 1223 836 1195 1339 664 318  
1225 1328 1082 548 132 746 776 193 397 945 1214 111 573 1153  
1400 541 432 288 669 530 632 1334 638 1093 1217 1353 120 682  
489 807 827 871 665 1040 1420 240 1280 534 1456 658 142 1127  
1031 1189 1354 1467 922 394 1312 750 441 684 249 841 147 528  
594 470 957 542 802 1355 1150 1329 959 1033 1316 364 438 689  
201 1427 857 933 1181 1395 662 1436 194 967 1496 1169 1145 630  
303 1256 440 1450 1452 465 702 1157 602 1480 1268 713 134 526  
1380 140 629 1356 328 1084 931 692 1069 313 894 556 1344 290  
138 926 1261 472 1002 878 905 1180 121 1136 635 1151 644 1045  
829 1242 1469 896 992 1052 1147 1396 663 119 979 319 1413 944  
1323 532 818 854 1034 771 1401 1431 976 1411 1300 252 1327 832  
1017 1199 504 505 916 1247 685 269 1416 833 307 1311 128 488  
529 1210 1463 675 1385 1403 452 666 1158 228 996 728 1315 322  
1479 797 1070 442 496 1372 920 688 1449 1117 636 506 444 950  
889 555 230 1232 566 1302 812 1476 218 1132 1105 906 849 390  
106 1249 192 553 117 185 1091 723 1220 588 1377 1018 1275 798  
672 1162 508 1482 559 210 928 1001 549 1124 738 570 1130 1192  
343 144 1296 1309 483 810 544 1062 1319 641 1332 756 845 593  
1171 350 921 1144 143 1046 575 156 1283 755 304 1178 329 1362  
1371 202 253 164 1107 759 1305 982 821 1381 480 1473 891 1063  
645 1490 317 422 1485 1368 1448 296 1398 1349 986 1099 1116 1499  
983 1009 1303 1274 1277 587 413 1276 988 1474 163 267 619 302  
443 828 561 426 232 1306 1094 509 775 195 258 471 799 956  
535 1495 446 1245 703 823 1246 622 1287 448 254 1365 538 525  
558 782 362 1236 1112 204 1343 604 1216 646 160 238 1397 306  
991 482 1176 913 1076 727 885 243 806 817 1410 1207 1442 693  
929 562 608 580 970 1179 294 314 316 654 168 381 217 501  
650 141 804 975 1090 346 430 268 167 621 527 883 954 310  
719 725 715 657 1146 182 376 571 384 791 1111 1243 1092 1325  
805 213 118 676 1252 286 1258 932 1041 859 720 946 1184 436  
589 760 887 1318 625 180 586 1012 661 930 342 1230 1271 1278  
607 130 300 583 1418 1269 379 395 1265 1222 341 868 1231 102  
881 1383 1075 374 1086 781 177 500 1425 1454 617 1085 995 1122  
618 546 462 1198 1272 154 1137 1188 188 1333 867 263 938 129  
616 498 1404 1053 289 1376 231 152 882 903 1379 335 722 461  
974 1126 840 1134 248 955 939 1391 1206 287 1441 109 1066 277  
1055 265 135 247 1035 266 145 1038 1234 1109 1089 788 124 660  
1186 1464 796 415 769 1003 1366 330 1492 1204 309 1330 469 697  
1262 1050 770 406 203 1308 984 439 793 1451 1182 174 490 718  
433 773 603 874 367 199 481 647 1384 902 819 862 1457 977  
942 1402 1421 1361 917 200 150 179 696 116 363 107 1465 458  
1212 1103 966 1010 326 1098 969 1167 694 1320 536 373 599 251  
131 237 1429 648 735 531 429 968 879 640 412 848 360 1138  
325 1322 299 1030 634 524 256 1060 935 495 282 206 943 523  
507 601 855 1291 1405 1369 999 1202 285 404 736 1498 1200 1439  
499 205 683 1462 949 652 332 1475 337 971 1174 667 560 172  
383 1255 359 401 377 592 1445 1221 866 981 447 1326 748 990  
405 115 790 830 1193 1423 467 271 410 1083 516 224 136 1029  
333 1440 674 1342 898 824 492 598 740 888 1288 104 1108 479  
1351 474 437 884 1370 264 1059 563 457 1313 241 1015 336 1387  
170 208 671 711 737 1470 365 763 567 486 772 301 311 584  
880 392 148 708 1259 786 370 678 146 581 918 1238 585 741

552 369 717 543 964 792 611 176 897 600 1054 428 181 211  
 1079 590 305 953 478 1375 244 511 1294 196 734 1239 1253 1128  
 1336 234 766 261 1194 431 572 1422 1297 574 355 207 706 280  
 726 414 352 1224 459 1254 1131 835 1172 1266 783 219 1213 1096  
 1251 1394 605 1064 1337 937 157 754 1168 155 1444 189 911 1321  
 1154 557 642 801 161 1382 1037 105 582 704 345 1120 1378 468  
 613 1023 628 466]  
 Department : ['Cardiology' 'Maternity' 'Neurology']  
 DistanceFromHome : [ 1 8 2 3 24 23 27 16 15 26 19 21 5 11 9 7 6 10 4 25 12 18 29 22  
 14 20 28 17 13]  
 Education : [2 1 4 3 5]  
 EducationField : ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'  
 'Human Resources']  
 EmployeeCount : [1]  
 EnvironmentSatisfaction : [2 3 4 1]  
 Gender : ['Female' 'Male']  
 HourlyRate : [ 94 61 92 56 40 79 81 67 44 84 49 31 93 50 51 80 96 78  
 45 82 53 83 58 72 48 42 41 86 97 75 33 37 73 98 36 47  
 71 30 43 99 59 95 57 76 87 66 55 32 52 70 62 64 63 60  
 100 46 39 77 35 91 54 34 90 65 88 85 89 68 69 74 38]  
 JobInvolvement : [3 2 4 1]  
 JobLevel : [2 1 3 4 5]  
 JobRole : ['Nurse' 'Other' 'Therapist' 'Administrative' 'Admin']  
 JobSatisfaction : [4 2 3 1]  
 MaritalStatus : ['Single' 'Married' 'Divorced']  
 MonthlyIncome : [ 5993 5130 2090 ... 12742 9071 2836]  
 MonthlyRate : [19479 24907 2396 ... 11563 11757 26997]  
 NumCompaniesWorked : [8 1 6 9 0 4 5 2 7 3]  
 Over18 : ['Y']  
 OverTime : [1 0]  
 PercentSalaryHike : [11 23 15 12 13 20 22 21 17 14 16 18 19 24 25]  
 PerformanceRating : [3 4]  
 RelationshipSatisfaction : [1 4 2 3]  
 StandardHours : [80]  
 Shift : [0 1 3 2]  
 TotalWorkingYears : [ 8 10 7 6 12 1 17 5 3 31 13 0 26 24 22 9 19 2 23 14 15 4 29 28  
 21 25 20 11 16 37 38 30 40 18 36 34 32 33 35 27]  
 TrainingTimesLastYear : [0 3 2 5 1 4 6]  
 WorkLifeBalance : [1 3 2 4]  
 YearsAtCompany : [ 6 10 0 8 2 7 1 9 5 4 25 3 12 14 22 15 27 21 17 11 13 37 16 20  
 40 24 33 19 36 18 29 31 32 34 26 30 23]  
 YearsInCurrentRole : [ 4 7 0 2 5 9 8 3 6 13 1 15 14 16 11 10 12 18 17]  
 YearsSinceLastPromotion : [ 0 1 3 2 7 4 8 6 5 15 9 13 12 10 11 14]  
 YearsWithCurrManager : [ 5 7 0 2 6 8 3 11 17 1 4 12 9 10 15 13 16 14]

```
In [ ]: 1 watson_data = pd.get_dummies(data = data, columns = ['Department','MaritalStatus','Busine
```

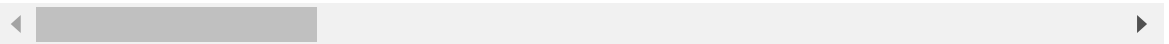
In []:

```
1 watson_data
```

Out[55]:

	Age	Attrition	DailyRate	DistanceFromHome	Education	EmployeeCount	Environment
0	41	0	1102	1	2	1	
1	49	0	279	8	1	1	
2	37	1	1373	2	2	1	
3	33	0	1392	3	4	1	
4	27	0	591	2	1	1	
...	...	...	...	...	...	...	...
1671	26	1	471	24	3	1	
1672	46	0	1125	10	3	1	
1673	20	0	959	1	3	1	
1674	39	0	466	1	1	1	
1675	27	0	511	2	2	1	

1676 rows × 49 columns



In []:

```
1 label_encoder = LabelEncoder()  
2 watson_data['Gender'] = label_encoder.fit_transform(watson_data['Gender'])
```

In []:

```
1 col_unique_value(watson_data)
```

Age : [41 49 37 33 27 32 59 30 38 36 35 29 31 34 28 22 53 24 21 42 44 46 39 43  
50 26 48 55 45 56 23 51 40 54 58 20 25 19 57 52 47 18 60]  
Attrition : [0 1]  
DailyRate : [1102 279 1373 1392 591 1005 1324 1358 216 1299 809 153 670 1346  
103 1389 334 1123 1219 371 673 1218 419 391 699 1282 1125 691  
477 705 924 1459 125 895 813 1273 869 890 852 1141 464 1240  
1357 994 721 1360 1065 408 1211 1229 626 1434 1488 1097 1443 515  
853 1142 655 1115 427 653 989 1435 1223 836 1195 1339 664 318  
1225 1328 1082 548 132 746 776 193 397 945 1214 111 573 1153  
1400 541 432 288 669 530 632 1334 638 1093 1217 1353 120 682  
489 807 827 871 665 1040 1420 240 1280 534 1456 658 142 1127  
1031 1189 1354 1467 922 394 1312 750 441 684 249 841 147 528  
594 470 957 542 802 1355 1150 1329 959 1033 1316 364 438 689  
201 1427 857 933 1181 1395 662 1436 194 967 1496 1169 1145 630  
303 1256 440 1450 1452 465 702 1157 602 1480 1268 713 134 526  
1380 140 629 1356 328 1084 931 692 1069 313 894 556 1344 290  
138 926 1261 472 1002 878 905 1180 121 1136 635 1151 644 1045  
829 1242 1469 896 992 1052 1147 1396 663 119 979 319 1413 944  
1323 532 818 854 1034 771 1401 1431 976 1411 1300 252 1327 832  
1017 1100 504 505 816 1217 685 360 1416 822 307 1211 120 480]

```
In [ ]: 1 col_to_scale = ["YearsWithCurrManager", "YearsSinceLastPromotion", "YearsInCurrentRole",  
2                 "TotalWorkingYears", "Shift", "RelationshipSatisfaction", "PercentSalaryHike", "NumCom  
3                 "JobInvolvement", "HourlyRate", "EnvironmentSatisfaction", "Education", "DistanceFro  
4 scaler = MinMaxScaler()  
5 watson_data[col_to_scale] = scaler.fit_transform(watson_data[col_to_scale])
```

```
In [ ]: 1 # col_unique_value(watson_data)
```

```
In [ ]: 1 watson_data.drop(["EmployeeCount", "Over18", "StandardHours"], axis = 1, inplace = True)
```

```
In [ ]: 1 # col_unique_value(watson_data)
```

```
In [ ]: 1 X = watson_data.drop(["Attrition"], axis = 1)
        2 y = watson_data['Attrition']
        3 y,X
```

```

Out[62]: (0    0
          1    0
          2    1
          3    0
          4    0
          ..
        1671   1
        1672   0
        1673   0
        1674   0
        1675   0
        Name: Attrition, Length: 1676, dtype: int64,
          Age DailyRate DistanceFromHome Education \
0    0.547619  0.715820    0.000000    0.25
1    0.738095  0.126700    0.250000    0.00
2    0.452381  0.909807    0.035714    0.25
3    0.357143  0.923407    0.071429    0.75
4    0.214286  0.350036    0.035714    0.00
...    ...    ...    ...    ...
1671 0.190476  0.264137    0.821429    0.50
1672 0.666667  0.732283    0.321429    0.50
1673 0.047619  0.613457    0.000000    0.50
1674 0.500000  0.260558    0.000000    0.00
1675 0.214286  0.292770    0.035714    0.25

          EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel \
0           0.333333    0  0.914286    0.666667    0.25
1           0.666667    1  0.442857    0.333333    0.25
2           1.000000    1  0.885714    0.333333    0.00
3           1.000000    0  0.371429    0.666667    0.00
4           0.000000    1  0.142857    0.666667    0.00
...    ...    ...    ...    ...
1671           0.666667    1  0.514286    0.000000    0.00
1672           0.666667    0  0.914286    0.333333    0.50
1673           1.000000    0  0.757143    0.333333    0.00
1674           1.000000    0  0.500000    0.333333    0.75
1675           0.000000    0  0.842857    1.000000    0.25

          JobSatisfaction ... JobRole_Administrative JobRole_Nurse \
0           1.000000 ...           0           1
1           0.333333 ...           0           0
2           0.666667 ...           0           1
3           0.666667 ...           0           0
4           0.333333 ...           0           1
...    ...    ...    ...
1671           1.000000 ...           0           1
1672           1.000000 ...           0           1
1673           0.333333 ...           0           0
1674           1.000000 ...           0           0
1675           0.666667 ...           0           1

          JobRole_Other JobRole_Therapist EducationField_Human Resources \
0           0           0           0
1           1           0           0
2           0           0           0
3           1           0           0
4           0           0           0
...    ...    ...
1671           0           0           0
1672           0           0           0
1673           1           0           0

```

1674	0	1	0
1675	0	0	0

	EducationField_Life Sciences	EducationField_Marketing \
0	1	0
1	1	0
2	0	0
3	1	0
4	0	0
...	...	...
1671	0	0
1672	0	1
1673	1	0
1674	1	0
1675	0	0

	EducationField_Medical	EducationField_Other \
0	0	0
1	0	0
2	0	1
3	0	0
4	1	0
...	...	...
1671	0	0
1672	0	0
1673	0	0
1674	0	0
1675	1	0

	EducationField_Technical Degree
0	0
1	0
2	0
3	0
4	0
...	...
1671	1
1672	0
1673	0
1674	0
1675	0

[1676 rows x 45 columns])

```
In [ ]: 1 X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.25)
```

```
In [ ]: 1 X_train.shape,y_train.shape , X_test.shape , y_test.shape
```

Out[64]: ((1257, 45), (1257, 45), (419, 45), (419, 45))

```
In [ ]: 1 model = keras.Sequential([
2         keras.layers.Dense(45,input_shape=(45,),activation="softmax"),
3         keras.layers.Dense(100,activation="relu"),
4         keras.layers.Dense(1,activation="sigmoid")])
5 model.compile(optimizer="adam",
6               loss = "binary_crossentropy",
7               metrics = [ "accuracy"])
```

```
In [ ]: 1 model.summary()
```

Model: "sequential\_5"

Layer (type)	Output Shape	Param #
dense_15 (Dense)	(None, 45)	2070
dense_16 (Dense)	(None, 100)	4600
dense_17 (Dense)	(None, 1)	101

Total params: 6771 (26.45 KB)  
 Trainable params: 6771 (26.45 KB)  
 Non-trainable params: 0 (0.00 Byte)

```
In [ ]: 1 model.fit(X_train,y_train,epochs=10)
```

```
Epoch 1/10
40/40 [=====] - 1s 2ms/step - loss: 0.5794 - accuracy: 0.8751
Epoch 2/10
40/40 [=====] - 0s 1ms/step - loss: 0.4150 - accuracy: 0.8751
Epoch 3/10
40/40 [=====] - 0s 1ms/step - loss: 0.3737 - accuracy: 0.8751
Epoch 4/10
40/40 [=====] - 0s 1ms/step - loss: 0.3668 - accuracy: 0.8751
Epoch 5/10
40/40 [=====] - 0s 1ms/step - loss: 0.3564 - accuracy: 0.8751
Epoch 6/10
40/40 [=====] - 0s 1ms/step - loss: 0.3362 - accuracy: 0.8751
Epoch 7/10
40/40 [=====] - 0s 1ms/step - loss: 0.3048 - accuracy: 0.8751
Epoch 8/10
40/40 [=====] - 0s 1ms/step - loss: 0.2694 - accuracy: 0.8751
Epoch 9/10
40/40 [=====] - 0s 1ms/step - loss: 0.2403 - accuracy: 0.8982
Epoch 10/10
40/40 [=====] - 0s 1ms/step - loss: 0.2215 - accuracy: 0.9117
```

Out[77]: <keras.src.callbacks.History at 0x7e90667b2830>

```
In [ ]: 1 model.evaluate(X_test,y_test)
```

```
14/14 [=====] - 0s 1ms/step - loss: 0.1929 - accuracy: 0.9308
```

Out[78]: [0.19290317595005035, 0.9307875633239746]



```
In []: 1 y_predict = model.predict(X_test)
```

14/14 [=====] - 0s 1ms/step

```
In []: 1 y_predict[4]
```

Out[80]: array([0.08099402], dtype=float32)

```
In []: 1 y_test[4]
```

Out[81]: 0

```
In []: 1 y_test[7],y_predict[7]
```

Out[82]: (0, array([0.01645084], dtype=float32))

```
In []: 1 cm= tf.math.confusion_matrix(y_test,y_predict)
```

```
In []: 1 cm
```

Out[87]: <tf.Tensor: shape=(2, 2), dtype=int32, numpy=  
array([[377, 0],  
 [ 42, 0]], dtype=int32)>

```
In []: 1 # test data is biased  
2 test_data = pd.read_csv("/content/drive/MyDrive/watson_shap_values_for_testset.csv")
```

```
In []: 1 test_data
```

Out[74]:

	EmployeeID	Age	BusinessTravel	DailyRate	Department	DistanceFromHome	Edu
0	10011	-0.082934	-0.049007	0.033110	-0.052101	-0.094783	-0.0
1	10012	-0.061724	-0.185156	0.004095	-0.055188	-0.063846	-0.0
2	10013	0.006087	-0.057510	0.017782	-0.002091	-0.008152	-0.0
3	10014	-0.012834	0.242323	-0.044950	-0.039829	-0.133996	-0.0
4	10015	-0.035391	-0.023601	0.014274	0.015462	-0.138073	0.0
...	...	...	...	...	...	...	...
201	10212	-0.039493	-0.034747	-0.044727	-0.022448	-0.117214	0.0
202	10213	-0.091602	-0.053394	-0.107608	-0.053622	-0.109185	-0.0
203	10214	0.249557	-0.104146	0.015778	-0.091577	0.014157	0.0
204	10215	-0.015204	-0.034563	-0.025504	-0.055249	-0.047021	0.0
205	10216	0.195810	-0.077439	0.048498	0.018811	0.056606	-0.0

206 rows × 33 columns



In [ ]: 1 y\_test\_data\_predict = model.predict(test\_data)

```
-----  
InvalidArgumentError Traceback (most recent call last)  
<ipython-input-84-0f8145c74e3d> in <cell line: 1>()  
----> 1 y_test_data_predict = model.predict(test_data)  
  
/usr/local/lib/python3.10/dist-packages/keras/src/utils/traceback_utils.py in error_handler(*args, **kwargs)  
    68     # To get the full stack trace, call:  
    69     # `tf.debugging.disable_traceback_filtering()`  
--> 70     raise e.with_traceback(filtered_tb) from None  
    71 finally:  
    72     del filtered_tb  
  
/usr/local/lib/python3.10/dist-packages/tensorflow/python/eager/execute.py in quick_execute(op_name, num_outputs, inputs, attrs, ctx, name)  
    51 try:  
    52     ctx.ensure_initialized()  
--> 53     tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,  
    54                                         inputs, attrs, num_outputs)  
-->
```

In [ ]: 1