In [2]:

```
import pandas as pd
import numpy as np
from sklearn import decomposition
import matplotlib.pyplot as plt
```

In [3]:

```
dataset = pd.read_csv("row_data.csv")
```

In [4]:

```
dataset
```

```
19988    64    85    84    73    88    66    81    85

19989    73    60    60    83    62    83    87    81

19990    71    83    82    94    91    67    86    76

19991    77    91    74    62    66    81    93    61



19992    79    62    86    87    93    81    64    68
```

Exploratory Data Analysis

In [651]:

```
dataset.describe()
```

Out[651]:

	Acedamic percentage in Operating Systems	percentage in Algorithms	Percentage in Programming Concepts	Percentage in Software Engineering	Percentage in Computer Networks	Percentage in Electronics Subjects	F in A
count	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20000.000000	20(
mean	77.002300	76.948200	77.017550	77.094500	76.958200	77.015550	
std	10.085697	10.101733	10.134815	10.087837	10.020088	10.168888	
min	60.000000	60.000000	60.000000	60.000000	60.000000	60.000000	
25%	68.000000	68.000000	68.000000	68.000000	68.000000	68.000000	
50%	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	
75%	86.000000	86.000000	86.000000	86.000000	85.000000	86.000000	
max	94.000000	94.000000	94.000000	94.000000	94.000000	94.000000	

In [652]:

```
data = dataset.iloc[:,:].values
label = dataset.iloc[:,-1].values
```

In [653]:

```
data
```

Out[653]:

```
array([[69, 63, 78, ..., 'yes', 'no', 'Database Developer'],
       [78, 62, 73, ..., 'no', 'yes', 'Portal Administrator'],
       [71, 86, 91, ..., 'no', 'yes', 'Portal Administrator'],
       ...,
       [83, 70, 80, ..., 'no', 'yes', 'Business Intelligence Analys
t'],
       [68, 87, 91, ..., 'yes', 'no',
        'Software Quality Assurance (QA)  Testing'],
       [73, 77, 74, ..., 'yes', 'no', 'Applications Developer']],
      dtype=object)
```

In [654]:

```
data.shape
```

Out[654]:

```
(20000, 39)
```

In [655]:

```
label.shape
```

Out[655]:

```
(20000,)
```

In [656]:

```
list(dataset.columns)
```

Out[656]:

```
['Acedamic percentage in Operating Systems',
 'percentage in Algorithms',
 'Percentage in Programming Concepts',
 'Percentage in Software Engineering',
 'Percentage in Computer Networks',
 'Percentage in Electronics Subjects',
 'Percentage in Computer Architecture',
 'Percentage in Mathematics',
 'Percentage in Communication skills',
 'Hours working per day',
 'Logical quotient rating',
 'hackathons',
 'coding skills rating',
 'public speaking points',
 'can work long time before system',
 'self-learning capability',
 'Extra-courses did',
 'certifications',
 'workshops',
 'talenttests taken',
 'olympiads',
 'reading and writing skills',
 'memory capability score',
 'Interested subjects',
 'interested career area ',
 'JobHigher Studies',
 'Type of company want to settle in',
 'Taken inputs from seniors or elders',
 'interested in games',
 'Interested Type of Books',
 'Salary Range Expected',
 'In a Realtionship',
 'Gentle or Tuff behaviour',
 'Management or Technical',
 'Salarywork',
 'hardsmart worker',
 'worked in teams ever',
 'Introvert',
 'Suggested Job Role']
```

In [657]:

```
data.size
```

Out[657]:

```
780000
```

In [658]:

```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

In [659]:

```
labelencoder = LabelEncoder()
```

In [660]:

```
for i in range(14,38):
    data[:,i] = labelencoder.fit_transform(data[:,i])
```

In [661]:

```
data
```

Out[661]:

```
array([[69, 63, 78, ..., 1, 0, 'Database Developer'],
       [78, 62, 73, ..., 0, 1, 'Portal Administrator'],
       [71, 86, 91, ..., 0, 1, 'Portal Administrator'],
       ...,
       [83, 70, 80, ..., 0, 1, 'Business Intelligence Analyst'],
       [68, 87, 91, ..., 1, 0,
        'Software Quality Assurance (QA)  Testing'],
       [73, 77, 74, ..., 1, 0, 'Applications Developer']], dtype=objec
t)
```

In [662]:

```
data.shape
```

Out[662]:

```
(20000, 39)
```

Subsets of Data

Academic data

In [663]:

```
academic_data_r=dataset.iloc[:,0:9]
```

In [664]:

```
academic_data_r.head()
```

Out[664]:

	Acedamic percentage in Operating Systems	percentage in Algorithms	Percentage in Programming Concepts	Percentage in Software Engineering	Percentage in Computer Networks	Percentage in Electronics Subjects	Percentage in Computer Architecture	I Ma
0	69	63	78	87	94	94	87	
1	78	62	73	60	71	70	73	
2	71	86	91	87	61	81	72	
3	76	87	60	84	89	73	62	
4	92	62	90	67	71	89	73	

In [665]:

```
academic_data=data[:,0:9]
```

In [666]:

```
academic_data
```

Out[666]:

```
array([[69, 63, 78, ..., 87, 84, 61],
       [78, 62, 73, ..., 73, 84, 91],
       [71, 86, 91, ..., 72, 72, 94],
       ...,
       [83, 70, 80, ..., 69, 94, 88],
       [68, 87, 91, ..., 61, 87, 61],
       [73, 77, 74, ..., 92, 73, 90]], dtype=object)
```

In [667]:

```
academic_data.shape
```

Out[667]:

```
(20000, 9)
```

In [668]:

```
np.sum(academic_data)
```

Out[668]:

```
13858814
```

In [669]:

```
 academic_percentage=np.sum(academic_data, axis = 1, keepdims = True)
```

In [670]:

```
academic_percentage
```

Out[670]:

```
array([[717],
       [662],
       [715],
       ...,
       [720],
       [683],
       [698]], dtype=object)
```

In [671]:

```
academic_percentage=np.true_divide(academic_percentage, 9)
```

In [672]:

```
academic_percentage
```

Out[672]:

```
array([[79.66666666666667],
       [73.55555555555556],
       [79.44444444444444],
       ...,
       [80.0],
       [75.88888888888889],
       [77.55555555555556]], dtype=object)
```

Communication Skills Data

In [673]:

```
communication_skill_data_r=dataset.iloc[:,[13,21]]
```

In [674]:

```
communication_skill_data=data[:,[13,21]]
```

In [675]:

```
communication_skill_data_r.head(n=10)
```

Out[675]:

	public speaking points	reading and writing skills
0	8	excellent
1	3	poor
2	3	poor
3	5	medium
4	3	poor
5	1	poor
6	3	excellent
7	6	poor
8	8	poor
9	4	excellent

In [676]:

```
communication_skill_data
```

Out[676]:

```
array([[8, 0],
       [3, 2],
       [3, 2],
       ...,
       [3, 1],
       [5, 2],
       [6, 0]], dtype=object)
```

In [677]:

```
communication_skill_data.shape
```

Out[677]:

```
(20000, 2)
```

In [678]:

```
np.amax(communication_skill_data, axis=None, out=None)
```

Out[678]:

```
9
```

In [679]:

```
communication_percentage=np.sum(communication_skill_data, axis = 1, keepdims = True
```

In [680]:

```
communication_percentage
```

Out[680]:

```
array([[8],
       [5],
       [5],
       ...,
       [4],
       [7],
       [6]], dtype=object)
```

In [681]:

```
communication_percentage=np.true_divide(communication_percentage,12)
```

In [682]:

```
communication_percentage
```

Out[682]:

```
array([[0.6666666666666666],
       [0.4166666666666667],
       [0.4166666666666667],
       ...,
       [0.3333333333333333],
       [0.5833333333333334],
       [0.5]], dtype=object)
```

In [683]:

```
communication_percentage=np.multiply(communication_percentage,100)
```

In [684]:

```
communication_percentage
```

Out[684]:

```
array([[66.66666666666666],
       [41.66666666666667],
       [41.66666666666667],
       ...,
       [33.33333333333333],
       [58.333333333333336],
       [50.0]], dtype=object)
```

Teamwork_data

In [685]:

```
teamwork_data_r=dataset.iloc[:,[27,28,32,33,35,36]]
```

In [686]:

```
teamwork_data_r.head()
```

Out[686]:

	Taken inputs from seniors or elders	interested in games	Gentle or Tuff behaviour	Management or Technical	hardsmart worker	worked in teams ever
0	no	no	stubborn	Management	hard worker	yes
1	yes	yes	gentle	Technical	hard worker	no
2	yes	yes	stubborn	Management	hard worker	no
3	no	no	gentle	Management	smart worker	yes
4	no	yes	stubborn	Management	hard worker	yes

In [687]:

```
teamwork_data=data[:,[27,28,32,33,35,36]]
```

In [688]:

```
teamwork_data
```

Out[688]:

```
array([[0, 0, 1, 0, 0, 1],
       [1, 1, 0, 1, 0, 0],
       [1, 1, 1, 0, 0, 0],
       ...,
       [1, 1, 0, 1, 0, 0],
       [1, 0, 0, 0, 1, 1],
       [1, 0, 0, 0, 0, 1]], dtype=object)
```

In [689]:

```
teamwork_data.shape
```

Out[689]:

```
(20000, 6)
```

In [690]:

```
teamwork_percentage=np.sum(teamwork_data, axis = 1, keepdims = True)
```

In [691]:

```
teamwork_percentage=np.true_divide(teamwork_percentage,6)
```

In [692]:

```
teamwork_percentage=np.multiply(teamwork_percentage,100)
```

In [693]:

```
teamwork_percentage
```

Out[693]:

```
array([[33.33333333333333],
       [50.0],
       [50.0],
       ...,
       [50.0],
       [50.0],
       [33.33333333333333]], dtype=object)
```

problem_solving_data

In [694]:

```
problem_solving_data_r=dataset.iloc[:,[10,11,12,19,22,20]]
```

In [695]:

```
problem_solving_data_r.head(n=10)
```

Out[695]:

	Logical quotient rating	hackathons	coding skills rating	talenttests taken	memory capability score	olympiads
0	4	0	4	no	excellent	yes
1	7	1	2	no	medium	no
2	1	4	1	no	excellent	yes
3	1	1	2	yes	excellent	no
4	5	4	6	no	excellent	no
5	5	3	8	no	medium	no
6	3	2	3	no	poor	no
7	2	1	6	no	excellent	yes
8	5	2	4	yes	poor	no
9	9	0	5	yes	poor	yes

In [696]:

```
problem_solving_data=data[:,[10,11,12,19,22,20]]
```

In [697]:

```
problem_solving_data
```

Out[697]:

```
array([[4, 0, 4, 0, 0, 1],
       [7, 1, 2, 0, 1, 0],
       [1, 4, 1, 0, 0, 1],
       ...,
       [3, 6, 2, 1, 0, 1],
       [1, 4, 9, 0, 2, 1],
       [3, 1, 7, 0, 0, 1]], dtype=object)
```

In [698]:

```
problem_solving_data.shape
```

Out[698]:

```
(20000, 6)
```

In [699]:

```
problem_solving_percentage=np.sum(problem_solving_data,axis=1,keepdims=True)
```

In [700]:

```
problem_solving_percentage
```

Out[700]:

```
array([[9],
       [11],
       [7],
       ...,
       [13],
       [17],
       [12]], dtype=object)
```

In [701]:

```
np.amax(problem_solving_data[:,2])
```

Out[701]:

```
9
```

In [702]:

```
problem_solving_percentage=np.true_divide(problem_solving_percentage,37)
```

In [703]:

```
problem_solving_percentage=np.multiply(problem_solving_percentage,100)
```

In [704]:

```
problem_solving_percentage
```

Out[704]:

```
array([[24.324324324324326],
       [29.72972972972973],
       [18.91891891891892],
       ...,
       [35.13513513513514],
       [45.94594594594595],
       [32.432432432432435]], dtype=object)
```

self_managment_data

In [705]:

```
self_managment_data_r=dataset.iloc[:,[14,15,29,31,33,34,37]]
```

In [706]:

```
self_managment_data_r.head()
```

Out[706]:

	can work long time before system	self-learning capability	Interested Type of Books	In a Realtionship	Management or Technical	Salarywork	Introvert
0	yes	yes	Prayer books	no	Management	salary	no
1	yes	no	Childrens	yes	Technical	salary	yes
2	yes	no	Travel	no	Management	work	yes
3	no	yes	Romance	yes	Management	work	yes
4	no	no	Cookbooks	no	Management	work	yes

In [707]:

```
self_managment_data=data[:,[14,15,29,31,33,34,37]]
```

In [708]:

```
self_managment_data
```

Out[708]:

```
array([[1, 1, 21, ..., 0, 0, 0],
       [1, 0, 5, ..., 1, 0, 1],
       [1, 0, 29, ..., 0, 1, 1],
       ...,
       [1, 1, 10, ..., 1, 1, 1],
       [0, 0, 29, ..., 0, 1, 0],
       [1, 1, 6, ..., 0, 1, 0]], dtype=object)
```

In [709]:

```
self_managment_data.shape
```

Out[709]:

```
(20000, 7)
```

In [710]:

```
np.amax(self_managment_data[:,2])
```

Out[710]:

```
30
```

In [711]:

```
np.amax(self_managment_data[:,4])
```

Out[711]:

```
1
```

In [712]:

```
np.amax(self_managment_data[:,5])
```

Out[712]:

```
1
```

In [713]:

```
self_managment_percentage=np.sum(self_managment_data,axis=1,keepdims=True)
```

In [714]:

```
self_managment_percentage
```

Out[714]:

```
array([[23],
       [9],
       [32],
       ...,
       [16],
       [30],
       [9]], dtype=object)
```

In [715]:

```
self_managment_percentage=np.true_divide(self_managment_percentage,36)
```

In [716]:

```
self_managment_percentage
```

Out[716]:

```
array([[0.6388888888888888],
       [0.25],
       [0.8888888888888888],
       ...,
       [0.4444444444444444],
       [0.8333333333333334],
       [0.25]], dtype=object)
```

In [717]:

```
self_managment_percentage=np.multiply(self_managment_percentage,100)
```

In [718]:

```
self_managment_percentage
```

Out[718]:

```
array([[63.888888888888886],
       [25.0],
       [88.88888888888889],
       ...,
       [44.44444444444444],
       [83.33333333333334],
       [25.0]], dtype=object)
```

In [ ]:

```

```

Knowledge_data

In [719]:

```
knowledge_data_r=dataset.iloc[:,[16,17,18,23]]
```

In [720]:

```
knowledge_data_r.head()
```

Out[720]:

	Extra-courses did	certifications	workshops	Interested subjects
0	yes	shell programming	cloud computing	cloud computing
1	yes	machine learning	database security	networks
2	yes	app development	web technologies	hacking
3	no	python	data science	networks
4	no	app development	cloud computing	Computer Architecture

In [721]:

```
knowledge_data=data[:,[16,17,18,23]]
```

In [722]:

```
knowledge_data
```

Out[722]:

```
array([[1, 8, 0, 4],
       [1, 5, 2, 7],
       [1, 0, 7, 6],
       ...,
       [1, 4, 2, 7],
       [0, 2, 0, 1],
       [1, 0, 2, 1]], dtype=object)
```

In [723]:

```
knowledge_data.shape
```

Out[723]:

```
(20000, 4)
```

In [724]:

```
np.amax(knowledge_data[:,1])
```

Out[724]:

```
8
```

In [725]:

```
np.amax(knowledge_data[:,2])
```

Out[725]:

```
7
```

In [726]:

```
np.amax(knowledge_data[:,3])
```

Out[726]:

```
9
```

In [727]:

```
knowledge_percentage=np.sum(knowledge_data,axis=1,keepdims=True)
```

In [728]:

```
knowledge_percentage
```

Out[728]:

```
array([[13],
       [15],
       [14],
       ...,
       [14],
       [3],
       [4]], dtype=object)
```

In [729]:

```
knowledge_percentage=np.true_divide(knowledge_percentage,26)
```

In [730]:

```
knowledge_percentage=np.multiply(knowledge_percentage,100)
```

In [731]:

```
knowledge_percentage
```

Out[731]:

```
array([[50.0],
       [57.692307692307686],
       [53.84615384615385],
       ...,
       [53.84615384615385],
       [11.538461538461538],
       [15.384615384615385]], dtype=object)
```

Interests_data

In [732]:

```
interests_data_r=dataset.iloc[:,[24,25,26,29,30]]
```

In [733]:

```
interests_data_r.head()
```

Out[733]:

	interested career area	JobHigher Studies	Type of company want to settle in	Interested Type of Books	Salary Range Expected
0	system developer	higherstudies	Web Services	Prayer books	salary
1	Business process analyst	job	SAaS services	Childrens	salary
2	developer	higherstudies	Sales and Marketing	Travel	Work
3	testing	higherstudies	Testing and Maintainance Services	Romance	Work
4	testing	higherstudies	product development	Cookbooks	salary

In [734]:

```
interests_data=data[:,[24,25,26,29,30]]
```

In [735]:

```
interests_data
```

Out[735]:

```
array([[4, 0, 8, 21, 1],
       [0, 1, 4, 5, 1],
       [2, 0, 5, 29, 0],
       ...,
       [1, 0, 4, 10, 0],
       [5, 1, 1, 29, 0],
       [5, 1, 2, 6, 0]], dtype=object)
```

In [736]:

```
interests_data.shape
```

Out[736]:

```
(20000, 5)
```

In [737]:

```
np.amax(interests_data[:,0])
```

Out[737]:

```
5
```

In [738]:

```
np.amax(interests_data[:,1])
```

Out[738]:

```
1
```

In [739]:

```
np.amax(interests_data[:,2])
```

Out[739]:

```
9
```

In [740]:

```
np.amax(interests_data[:,3])
```

Out[740]:

```
30
```

In [741]:

```
np.amax(interests_data[:,4])
```

Out[741]:

```
1
```

In [742]:

```
interests_percentage=np.sum(interests_data,axis=1, keepdims= True)
```

In [743]:

```
interests_percentage
```

Out[743]:

```
array([[34],
       [11],
       [36],
       ...,
       [15],
       [36],
       [14]], dtype=object)
```

In [744]:

```
interests_percentage=np.true_divide(interests_percentage,48)
```

In [745]:

```
interests_percentage=np.multiply(interests_percentage,100)
```

In [746]:

```
interests_percentage
```

Out[746]:

```
array([[70.83333333333334],
       [22.916666666666664],
       [75.0],
       ...,
       [31.25],
       [75.0],
       [29.166666666666668]], dtype=object)
```

Concatenate arrays

In [747]:

```
combine_data=np.concatenate((academic_data,academic_percentage,communication_percent
```

In [748]:

```
combine_data.shape
```

Out[748]:

```
(20000, 16)
```

In [749]:

```
X1 = pd.DataFrame(combine_data,columns=['Acedamic percentage in Operating Systems',
                                        'percentage in Algorithms',
                                        'Percentage in Programming Concepts',
                                        'Percentage in Software Engineering',
                                        'Percentage in Computer Networks',
                                        'Percentage in Electronics Subjects',
                                        'Percentage in Computer Architecture',
                                        'Percentage in Mathematics',
                                        'Percentage in Communication skills',
                                        'academic_percentage',
                                        'communication_percentage',
                                        'teamwork_percentage',
                                        'problem_solving_percentage',
                                        'self_managment_percentage',
                                        'knowledge_percentage',
                                        'interests_percentage'])
```

In [750]:

```
X1.head()
```

Out[750]:

	Acedamic percentage in Operating Systems	percentage in Algorithms	Percentage in Programming Concepts	Percentage in Software Engineering	Percentage in Computer Networks	Percentage in Electronics Subjects	Percentage in Computer Architecture	P M
0	69	63	78	87	94	94	87	
1	78	62	73	60	71	70	73	
2	71	86	91	87	61	81	72	
3	76	87	60	84	89	73	62	
4	92	62	90	67	71	89	73	

In [751]:

```
label
```

Out[751]:

```
array(['Database Developer', 'Portal Administrator',
       'Portal Administrator', ..., 'Business Intelligence Analyst',
       'Software Quality Assurance (QA)  Testing',
       'Applications Developer'], dtype=object)
```

In [ ]:

```

```

In [752]:

```
label = labelencoder.fit_transform(label)
```

In [753]:

```
label
```

Out[753]:

```
array([ 7, 18, 18, ...,  1, 24,  0])
```

In [754]:

```
y=pd.DataFrame(label,columns=["Suggested Job Role"])
```

In [755]:

```
final_df = pd.concat((X1,y),axis=1)
```

In [756]:

```
final_df.head()
```

Out[756]:

	Acedamic percentage in Operating Systems	percentage in Algorithms	Percentage in Programming Concepts	Percentage in Software Engineering	Percentage in Computer Networks	Percentage in Electronics Subjects	Percentage in Computer Architecture
0	69	63	78	87	94	94	87
1	78	62	73	60	71	70	73
2	71	86	91	87	61	81	72
3	76	87	60	84	89	73	62
4	92	62	90	67	71	89	73

In [757]:

```
X.shape
```

Out[757]:

```
(20000, 16)
```

In [758]:

```
X
```

Out[758]:

```
array([[69, 63, 78, ..., 63.888888888888886, 50.0, 70.83333333333334],
       [78, 62, 73, ..., 25.0, 57.692307692307686, 22.91666666666666
4],
       [71, 86, 91, ..., 88.88888888888889, 53.84615384615385, 75.0],
       ...,
       [83, 70, 80, ..., 44.44444444444444, 53.84615384615385, 31.25],
       [68, 87, 91, ..., 83.33333333333334, 11.538461538461538, 75.0],
       [73, 77, 74, ..., 25.0, 15.384615384615385, 29.16666666666666
8]],
      dtype=object)
```

In [759]:

```
Y=label
```

In [760]:

```
Y.shape
```

Out[760]:

```
(20000,)
```

In [761]:

```
Y
```

Out[761]:

```
array([ 7, 18, 18, ...,  1, 24,  0])
```

In [762]:

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
```

In [763]:

```
test = SelectKBest(score_func=chi2, k=16)
fit = test.fit(X, Y)
```

In [764]:

```
np.set_printoptions(suppress=True)
print(fit.scores_)
```

```
[ 18.38034544  26.26991883  51.5398971   59.78959158  38.37664406
  56.65610078  23.31612484  31.38669513  31.52321969   3.66516634
 447.14703593 417.61605401 159.43569869 393.02932584 151.63485915
 252.38229034]
```

In [765]:

```
from sklearn import tree
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.metrics import accuracy_score
```

In [766]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=10)
```

In [767]:

```
clf = tree.DecisionTreeClassifier()
```

In [768]:

```
clf = clf.fit(X_train, y_train)
```

In [769]:

```
from sklearn.metrics import confusion_matrix,accuracy_score
```

In [770]:

```
y_pred = clf.predict(X_test)
```

In [771]:

```
y_pred
```

Out[771]:

```
array([31, 17, 15, ..., 23,  2, 21])
```

In [772]:

```
cm = confusion_matrix(y_test,y_pred)
accuracy = accuracy_score(y_test,y_pred)
```

In [773]:

```
print("confusion matrics=",cm)
print("  ")
print("accuracy=",accuracy*100)
```

```
confusion matrics= [[4 5 3 ... 5 2 5]
 [2 1 3 ... 3 1 6]
 [3 1 5 ... 5 1 6]
 ...
 [3 5 2 ... 2 2 1]
 [4 3 1 ... 4 4 2]
 [5 4 2 ... 4 2 3]]

accuracy= 3.2
```

In [797]:

```
final_df.head(n=10)
```

Out[797]:

	Acedamic percentage in Operating Systems	percentage in Algorithms	Percentage in Programming Concepts	Percentage in Software Engineering	Percentage in Computer Networks	Percentage in Electronics Subjects	Percentage in Computer Architecture
0	69	63	78	87	94	94	87
1	78	62	73	60	71	70	73
2	71	86	91	87	61	81	72
3	76	87	60	84	89	73	62
4	92	62	90	67	71	89	73
5	88	86	62	79	93	84	69
6	93	77	69	79	90	93	73
7	84	72	88	62	66	63	78
8	73	66	66	81	81	69	61
9	62	76	85	91	82	69	63

In [775]:

```
test=X[0,:]
```

In [776]:

```
test
```

Out[776]:

```
array([69, 63, 78, 87, 94, 94, 87, 84, 61, 79.66666666666667,
       66.66666666666666, 33.33333333333333, 24.324324324324326,
       63.888888888888886, 50.0, 70.83333333333334], dtype=object)
```

In [777]:

```
X_test.dtype
```

Out[777]:

```
dtype('O')
```

In [778]:

```
test=test.reshape(1, -1)
```

In [779]:

```
pred = clf.predict(test)
```

In [780]:

```
pred
```

Out[780]:

```
array([7])
```

In [781]:

```
label1 = dataset.iloc[:,-1].values
```

In [782]:

```
label1=pd.DataFrame(label1,columns=["Suggested Job Role "])
```

In [783]:

```
label2=pd.DataFrame(label,columns=["Suggested Job Role map"])
```

In [784]:

```
label_map=pd.concat((label1,label2),axis=1)
```

In [785]:

```
label_map.head()
```

Out[785]:

	Suggested Job Role	Suggested Job Role map
0	Database Developer	7
1	Portal Administrator	18
2	Portal Administrator	18
3	Systems Security Administrator	28
4	Business Systems Analyst	2

In [786]:

```
label_map=label_map.drop_duplicates()
```

In [791]:

```
label_map=label_map.sort_values('Suggested Job Role map')
```

In [792]:

```
label_map
```

Out[792]:

	Suggested Job Role	Suggested Job Role map
52	Applications Developer	0
7	Business Intelligence Analyst	1
4	Business Systems Analyst	2
18	CRM Business Analyst	3
9	CRM Technical Developer	4
39	Data Architect	5
57	Database Administrator	6
0	Database Developer	7
47	Database Manager	8
34	Design & UX	9
42	E-Commerce Analyst	10
17	Information Security Analyst	11
45	Information Technology Auditor	12
28	Information Technology Manager	13
10	Mobile Applications Developer	14
58	Network Engineer	15
38	Network Security Administrator	16
79	Network Security Engineer	17
1	Portal Administrator	18
30	Programmer Analyst	19
27	Project Manager	20
14	Quality Assurance Associate	21
41	Software Developer	22
70	Software Engineer	23
103	Software Quality Assurance (QA) Testing	24
5	Software Systems Engineer	25
35	Solutions Architect	26
37	Systems Analyst	27
3	Systems Security Administrator	28
77	Technical Engineer	29
43	Technical ServicesHelp DeskTech Support	30
23	Technical Support	31
11	UX Designer	32
15	Web Developer	33

In [800]:

```
test2=X[0:10,:]
```

In [801]:

```
test2
```

Out[801]:

```
array([[69, 63, 78, 87, 94, 94, 87, 84, 61, 79.66666666666667,
        66.66666666666666, 33.33333333333333, 24.324324324324326,
        63.888888888888886, 50.0, 70.83333333333334],
       [78, 62, 73, 60, 71, 70, 73, 84, 91, 73.55555555555556,
        41.66666666666667, 50.0, 29.72972972972973, 25.0,
        57.692307692307686, 22.916666666666664],
       [71, 86, 91, 87, 61, 81, 72, 72, 94, 79.44444444444444,
        41.66666666666667, 50.0, 18.91891891891892, 88.88888888888889,
        53.84615384615385, 75.0],
       [76, 87, 60, 84, 89, 73, 62, 88, 69, 76.44444444444444, 50.0,
        33.33333333333333, 13.513513513513514, 75.0, 53.8461538461538
5,
        72.91666666666666],
       [92, 62, 90, 67, 71, 89, 73, 71, 73, 76.44444444444444,
        41.66666666666667, 50.0, 40.54054054054054, 25.0, 0.0,
        45.83333333333333],
       [88, 86, 62, 79, 93, 84, 69, 71, 82, 79.33333333333333, 25.0,
        83.33333333333334, 45.94594594594595, 83.33333333333334,
        65.38461538461539, 83.33333333333334],
       [93, 77, 69, 79, 90, 93, 73, 63, 77, 79.33333333333333, 25.0,
        33.33333333333333, 27.027027027027028, 36.11111111111111,
        61.53846153846154, 29.166666666666668],
       [84, 72, 88, 62, 66, 63, 78, 94, 60, 74.11111111111111,
        66.66666666666666, 50.0, 27.027027027027028, 69.4444444444444
4,
        57.692307692307686, 54.166666666666664],
       [73, 66, 66, 81, 81, 69, 61, 87, 90, 74.88888888888889,
        83.33333333333334, 33.33333333333333, 37.83783783783784,
        61.111111111111114, 42.30769230769231, 41.66666666666667],
       [62, 76, 85, 91, 82, 69, 63, 63, 81, 74.66666666666667,
        33.33333333333333, 16.666666666666664, 48.64864864864865,
        63.888888888888886, 26.923076923076923, 66.66666666666666]],
      dtype=object)
```

In [805]:

```
test2_test=label[0:10]
```

In [806]:

```
test2_test
```

Out[806]:

```
array([ 7, 18, 18, 28,  2, 25,  7,  1,  2,  4])
```

In [802]:

```
pred = clf.predict(test2)
```

In [803]:

```
pred
```

Out[803]:

```
array([ 7, 18, 18, 28,  2, 25,  7, 12,  2,  6])
```

In [807]:

```
accuracy = accuracy_score(test2_test,pred)
```

In [809]:

```
accuracy*100
```

Out[809]:

```
80.0
```

In [ ]:

```

```

In [ ]:

```

```

In [811]:

```
## Testing xgb
```

In [815]:

```
y
```

Out[815]:

	Suggested Job Role
0	7
1	18
2	18
3	28
4	2
5	25
6	7
7	1
8	2
9	4
10	14
11	32
12	4
13	2
14	21
15	33
16	33
17	11
18	3
19	11
20	7
21	2
22	21
23	31
24	31
25	11
26	1
27	20
28	13
29	33
...	...
19970	1
19971	14
19972	19

	Suggested Job Role
19973	13
19974	10
19975	22
19976	20
19977	18
19978	23
19979	2
19980	23
19981	0
19982	11
19983	8
19984	29
19985	26
19986	13
19987	1
19988	33
19989	19
19990	33
19991	13
19992	31
19993	25
19994	23
19995	29
19996	10
19997	1
19998	24
19999	0

20000 rows × 1 columns

In [816]:

```
label
```

Out[816]:

```
array([ 7, 18, 18, ...,  1, 24,  0])
```

In [817]:

```
X_train,X_test,y_train,y_test=train_test_split(X,label,test_size=0.3,random_state=1
```

In [818]:

```
X_train.shape
X_test[0]
```

Out[818]:

```
array([89, 75, 73, 81, 72, 65, 81, 72, 92, 77.77777777777777,
       8.333333333333332, 66.66666666666666, 51.35135135135135,
       83.33333333333334, 53.84615384615385, 62.5], dtype=object)
```

In [819]:

```
X_train=pd.to_numeric(X_train[:,:].flatten())
```

In [820]:

```
X_train=X_train.reshape((14000,16))
```

In [ ]: