CSE 643: Artificial Intelligence
Assignment 4

Name: Harsh Kumar Roll Number: 2019043

Preprocessing Steps

- For categorical attributes, we first convert them to numeric values. We use the following map for conversion:

```
conversion_map = {
   'Introvert': {'yes': 1, 'no': 0}, 'worked in teams ever?': {'yes': 1, 'no': 0},
   'hard/smart worker': {'hard worker': 1, 'smart worker': 0},
   'Salary/work': {'salary': 1, 'work':0},
   'Management or Technical': {'Management': 1, 'Technical': 0},
   'Gentle or Tuff behaviour?': {'gentle': 1, 'stubborn': 0},
   'Salary Range Expected': {'salary': 1, 'Work': 0},
   'interested in games': {'yes': 1, 'no':0}, 'In a Realtionship?': {'yes': 1, 'no':0},
   'Taken inputs from seniors or elders': {'yes': 1, 'no': 0},
   'Job/Higher Studies?': {'job': 1, 'higherstudies': 0},
   'memory capability score': {'excellent': 1, 'medium': 0,'poor': -1},
   'reading and writing skills': {'excellent': 1, 'medium': 0,'poor': -1},
   'olympiads': {'yes':1, 'no': 0},'self-learning capability?': {'yes': 1, 'no': 0},
   'talenttests taken?': {'yes': 1, 'no': 0},'Extra-courses did': {'yes': 1, 'no': 0},
   'can work long time before system?':{'yes': 1, 'no': 0}
}
```

- For grades in courses, we scale down the grades to a smaller scale. 0 means a bad grade (below 34), 1 means average grade (34 to 68) and 68 above are considered to be good grades. We do this by dividing the course grade by 34, and converting the result into integer.

- We drop the ‘Suggested Job Role’ column, as it serves as the target class.

Reducing Classification Classes

We reduce the classification classes to fewer classes using the following map:

```
y_map = {
   'Database Developer': 'Database',
   'Portal Administrator': 'Administrator',
   'Systems Security Administrator': 'Administrator',
   'Business Systems Analyst': 'Analyst',
   'Software Systems Engineer': 'Engineer',
   'Business Intelligence Analyst': 'Analyst',
   'CRM Technical Developer': 'Developer',
   'Mobile Applications Developer': 'Developer',
   'UX Designer': 'Designer',
```

```
    'Quality Assurance Associate': 'QA',
    'Web Developer': 'Developer',
    'Information Security Analyst': 'Analyst',
    'CRM Business Analyst': 'Analyst',
    'Technical Support': 'Support',
    'Project Manager': 'Manager',
    'Information Technology Manager': 'Manager',
    'Programmer Analyst': 'Analyst',
    'Design & UX': 'Designer',
    'Solutions Architect': 'Engineer',
    'Systems Analyst': 'Analyst',
    'Network Security Administrator': 'Administrator',
    'Data Architect': 'Database',
    'Software Developer': 'Developer',
    'E-Commerce Analyst': 'Analyst',
    'Technical Services/Help Desk/Tech Support': 'Support',
    'Information Technology Auditor': 'Auditor',
    'Database Manager': 'Database',
    'Applications Developer': 'Developer',
    'Database Administrator': 'Database',
    'Network Engineer': 'Engineer',
    'Software Engineer': 'Egineer',
    'Technical Engineer': 'Engineer',
    'Network Security Engineer': 'Engineer',
    'Software Quality Assurance (QA) / Testing': 'QA'
}
```

This helps us make our classification task easier, by reducing redundant classes.
We then encode the labels, and prepare our data for feeding into the model.

Experimentation

We try different hyperparameters like the hidden layer size, activation function to be used and the learning_rate. I got the best results at the following values out of all possible values I tried:

```
clf = MLPClassifier(
    hidden_layer_sizes=(50, 50), activation='identity',
    max_iter = 300, learning_rate_init = 0.01, random_state = 0,
    verbose = True)
```

Values tried for hidden layer_sizes: (50), (100, 100), (200, 200)
Values tried for activation: relu, identity, logistic
Values tried for learning_rate_init: 0.01, 0.05, 0.1

Train / Test split

I tried three variants of train/test data split. 60-40 split for train and test respectively, 80-20, and 90-10. Though 90-10 split was giving the best accuracy. However, we have very few test cases

in the case of a 90-10 split. However, we don't gain much on the accuracy side by using 90-10 split, as compared to 80-20 split. Thus, I preferred the 80-20 split. 60-40 split is not a good way of splitting data into train and test sets, as we should have more data to train on to learn the pattern.

Results

Accuracy: 19.1%

	precision	recall	f1-score	support
0	0.08	0.00	0.01	477
1	0.20	0.85	0.32	772
2	0.00	0.00	0.00	112
3	0.00	0.00	0.00	463
4	0.00	0.00	0.00	211
5	0.14	0.07	0.09	583
6	0.00	0.00	0.00	103
7	0.19	0.11	0.14	592
8	0.00	0.00	0.00	230
9	0.00	0.00	0.00	241
10	0.00	0.00	0.00	216
accuracy			0.19	4000
macro avg	0.06	0.09	0.05	4000
weighted avg	0.10	0.19	0.10	4000

Confusion Matrix: