Chaithanya Chikkannaswamy

SUID : 387781563

Evolutionary Machine Learning – HW4

HW4: Use a Learning Classifier System for learning a collection of rules for the same classification problem. Compare with the previous results (HW1-3).

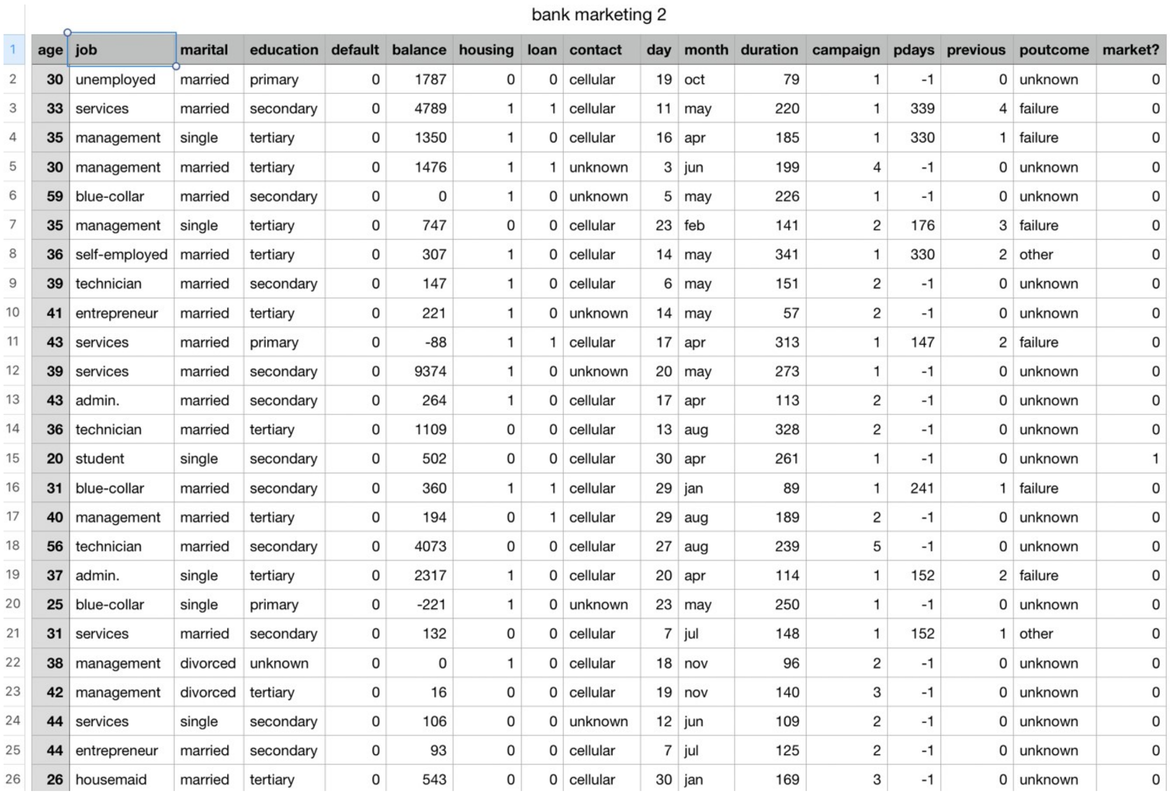
Link:
https://www.kaggle.com/chaithanya96/bankmarketing

The bank marketing dataset is the csv file with users' details used to predict the marketing decision. The decision yes or no is represented with binary numbers 1 and 0. The sample columns included are:

1. 'age',
2. 'job',
3. 'marital',
4. 'education',
5. 'default',
6. 'balance',
7. 'housing',
8. 'loan',
9. 'contact',
10. 'day',
11. 'month',
12. 'duration',
13. 'campaign',
14. 'pdays',
15. 'previous',
16. 'poutcome',
17. 'market?'

The screenshot of the sample dataset is attached below:

Chaithanya Chikkannaswamy



bank marketing 2

1	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	market?
2	30	unemployed	married	primary	0	1787	0	0	cellular	19	oct	79	1	-1	0	unknown	0
3	33	services	married	secondary	0	4789	1	1	cellular	11	may	220	1	339	4	failure	0
4	35	management	single	tertiary	0	1350	1	0	cellular	16	apr	185	1	330	1	failure	0
5	30	management	married	tertiary	0	1476	1	1	unknown	3	jun	199	4	-1	0	unknown	0
6	59	blue-collar	married	secondary	0	0	1	0	unknown	5	may	226	1	-1	0	unknown	0
7	35	management	single	tertiary	0	747	0	0	cellular	23	feb	141	2	176	3	failure	0
8	36	self-employed	married	tertiary	0	307	1	0	cellular	14	may	341	1	330	2	other	0
9	39	technician	married	secondary	0	147	1	0	cellular	6	may	151	2	-1	0	unknown	0
10	41	entrepreneur	married	tertiary	0	221	1	0	unknown	14	may	57	2	-1	0	unknown	0
11	43	services	married	primary	0	-88	1	1	cellular	17	apr	313	1	147	2	failure	0
12	39	services	married	secondary	0	9374	1	0	unknown	20	may	273	1	-1	0	unknown	0
13	43	admin.	married	secondary	0	264	1	0	cellular	17	apr	113	2	-1	0	unknown	0
14	36	technician	married	tertiary	0	1109	0	0	cellular	13	aug	328	2	-1	0	unknown	0
15	20	student	single	secondary	0	502	0	0	cellular	30	apr	261	1	-1	0	unknown	1
16	31	blue-collar	married	secondary	0	360	1	1	cellular	29	jan	89	1	241	1	failure	0
17	40	management	married	tertiary	0	194	0	1	cellular	29	aug	189	2	-1	0	unknown	0
18	56	technician	married	secondary	0	4073	0	0	cellular	27	aug	239	5	-1	0	unknown	0
19	37	admin.	single	tertiary	0	2317	1	0	cellular	20	apr	114	1	152	2	failure	0
20	25	blue-collar	single	primary	0	-221	1	0	unknown	23	may	250	1	-1	0	unknown	0
21	31	services	married	secondary	0	132	0	0	cellular	7	jul	148	1	152	1	other	0
22	38	management	divorced	unknown	0	0	1	0	cellular	18	nov	96	2	-1	0	unknown	0
23	42	management	divorced	tertiary	0	16	0	0	cellular	19	nov	140	3	-1	0	unknown	0
24	44	services	single	secondary	0	106	0	0	unknown	12	jun	109	2	-1	0	unknown	0
25	44	entrepreneur	married	secondary	0	93	0	0	cellular	7	jul	125	2	-1	0	unknown	0
26	26	housemaid	married	tertiary	0	543	0	0	cellular	30	jan	169	3	-1	0	unknown	0

Code execution steps:

Learning Classifier System

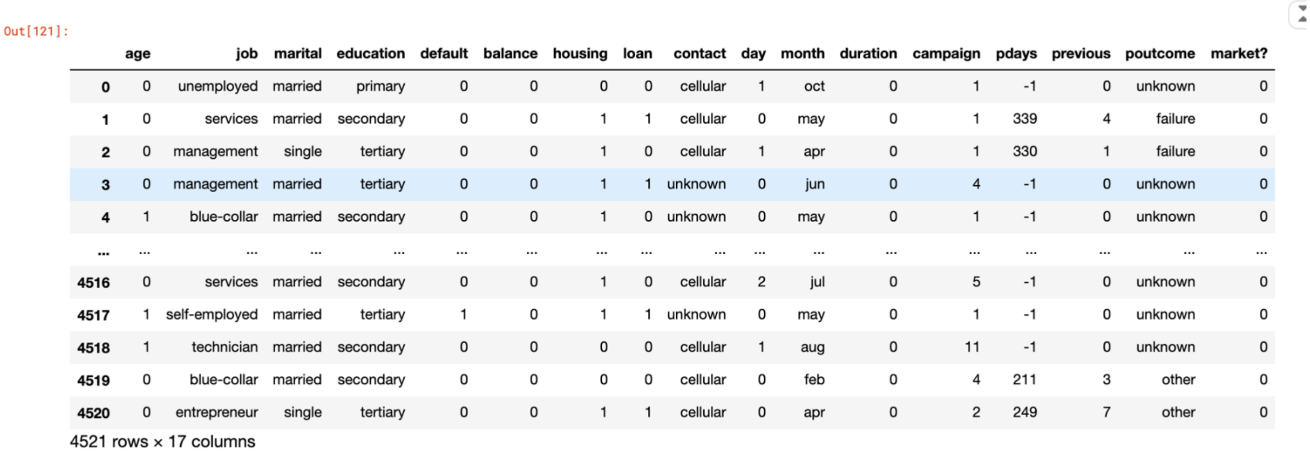
I have implemented the Learning Classifier System using XCS algorithm to train a shallow feedforward neural network for a 2-class classification task using the xcs library available in python. The accuracy is then compared with the accuracy calculated using Genetic Algorithm, CMA Evolutionary Strategy and particle swarm optimization implemented in previous HWs.

1. XCS python library is installed to implement the Learning Classifier System approach to train the data.
2. Pandas data frame available in python is used to read the data set.
3. The input data set is the continuous data which is converted to discreet data and then into binary data.
4. BankDataset Class is defined which takes the number of training cycles, input size and the input array as parameters.
5. XCS algorithm is applied on the dataset by specifying the default parameters. Few of the parameters are modified to obtain the efficient result. The parameters are modified as shown below:
    - algorithm.ga_threshold = 1
    - algorithm.crossover_probability = .5

- algorithm.wildcard_probability = .5
- algorithm.deletion_threshold = 2
- algorithm.subsumption_threshold = 10
- algorithm.mutation_probability = .03

6. The prediction accuracy is calculated for the model based on each column.

The snapshot of data set after converting it to discreet data:

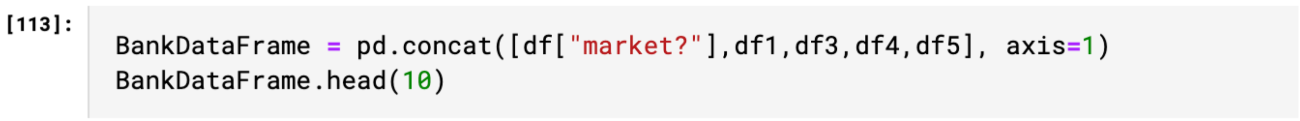


Out[121]:

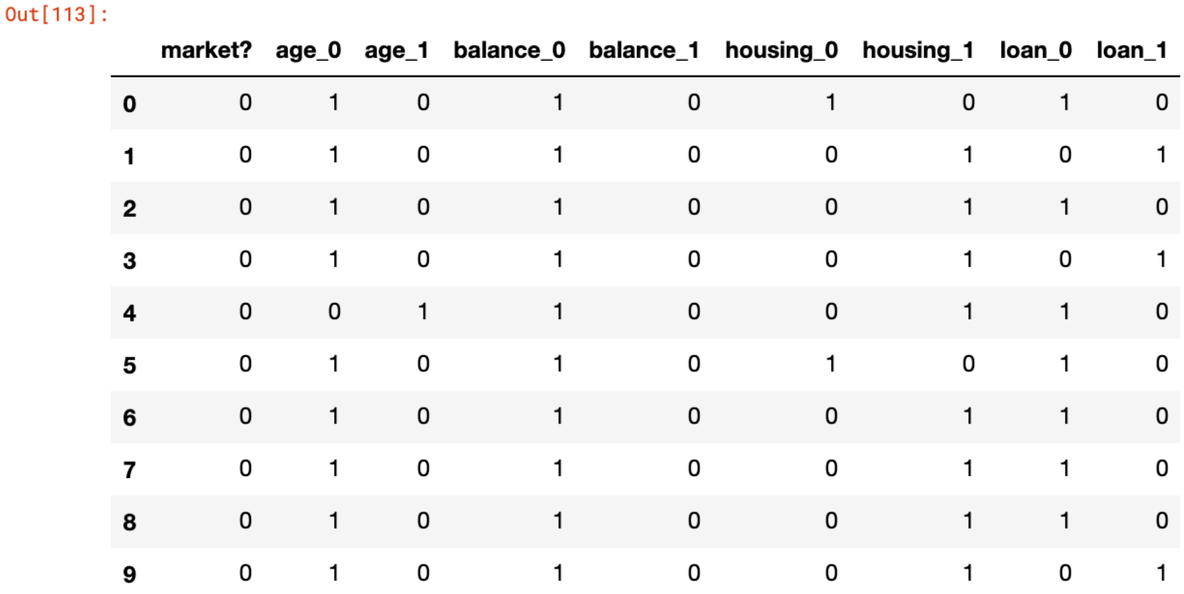
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	market?
0	0	unemployed	married	primary	0	0	0	0	cellular	1	oct	0	1	-1	0	unknown	0
1	0	services	married	secondary	0	0	1	1	cellular	0	may	0	1	339	4	failure	0
2	0	management	single	tertiary	0	0	1	0	cellular	1	apr	0	1	330	1	failure	0
3	0	management	married	tertiary	0	0	1	1	unknown	0	jun	0	4	-1	0	unknown	0
4	1	blue-collar	married	secondary	0	0	1	0	unknown	0	may	0	1	-1	0	unknown	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	0	services	married	secondary	0	0	1	0	cellular	2	jul	0	5	-1	0	unknown	0
4517	1	self-employed	married	tertiary	1	0	1	1	unknown	0	may	0	1	-1	0	unknown	0
4518	1	technician	married	secondary	0	0	0	0	cellular	1	aug	0	11	-1	0	unknown	0
4519	0	blue-collar	married	secondary	0	0	0	0	cellular	0	feb	0	4	211	3	other	0
4520	0	entrepreneur	single	tertiary	0	0	1	1	cellular	0	apr	0	2	249	7	other	0

4521 rows × 17 columns

The snapshot of data set after converting it to binary data:



```
[113]:  BankDataFrame = pd.concat([df["market?"],df1,df3,df4,df5], axis=1)
        BankDataFrame.head(10)
```

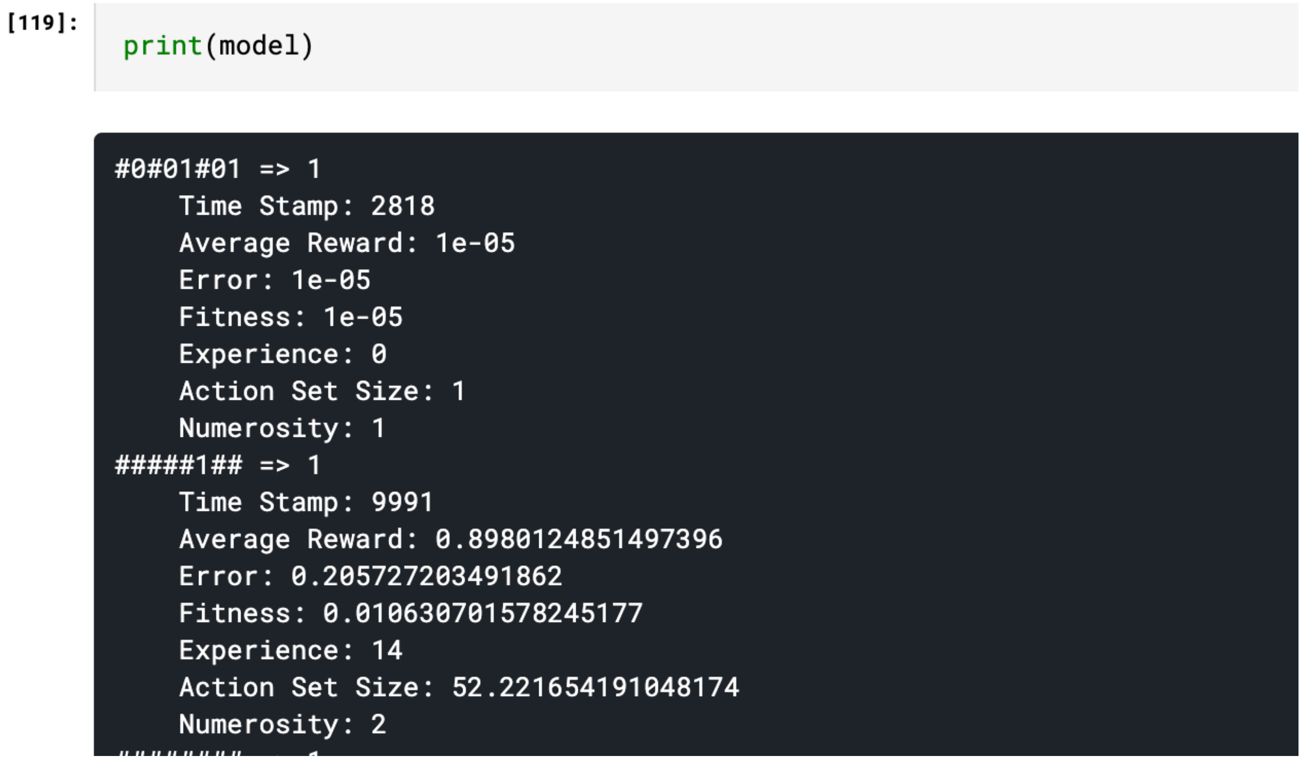


Out[113]:

	market?	age_0	age_1	balance_0	balance_1	housing_0	housing_1	loan_0	loan_1
0	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	0	1	0	1
2	0	1	0	1	0	0	1	1	0
3	0	1	0	1	0	0	1	0	1
4	0	0	1	1	0	0	1	1	0
5	0	1	0	1	0	1	0	1	0
6	0	1	0	1	0	0	1	1	0
7	0	1	0	1	0	0	1	1	0
8	0	1	0	1	0	0	1	1	0
9	0	1	0	1	0	0	1	0	1

Chaithanya Chikkannaswamy

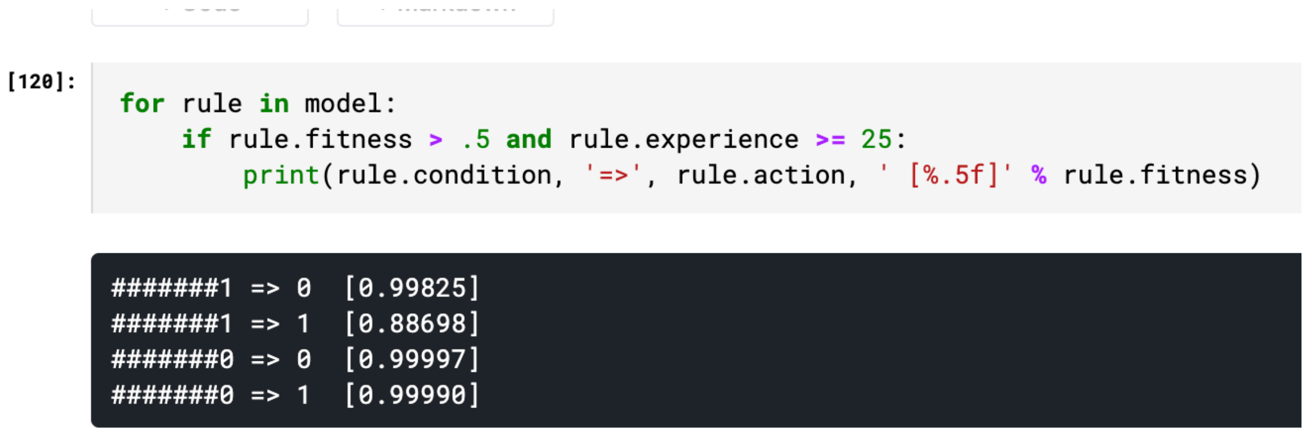
The snapshot of the model summary:



```
[119]:  print(model)
```

```
#0#01#01 => 1
    Time Stamp: 2818
    Average Reward: 1e-05
    Error: 1e-05
    Fitness: 1e-05
    Experience: 0
    Action Set Size: 1
    Numerosity: 1
#####1## => 1
    Time Stamp: 9991
    Average Reward: 0.8980124851497396
    Error: 0.205727203491862
    Fitness: 0.010630701578245177
    Experience: 14
    Action Set Size: 52.221654191048174
    Numerosity: 2
```

The snapshot of the predicted model accuracy:



```
[120]:  for rule in model:
            if rule.fitness > .5 and rule.experience >= 25:
                print(rule.condition, '=>', rule.action, ' [%.5f]' % rule.fitness)
```

```
#######1 => 0  [0.99825]
#######1 => 1  [0.88698]
#######0 => 0  [0.99997]
#######0 => 1  [0.99990]
```

Chaithanya Chikkannaswamy

Conclusion:

The **Learning Classifier System** resulted in the accuracy of 99%, 88%, 99% and 99% based on the columns. **Particle Swarm Optimization** resulted in the accuracy of 0.88 i.e. **88**%. **CMA-ES** resulted in 0.715 test accuracy. **Genetic Neural Network** resulted in a test accuracy of 0.89 and the **Sequential Neural Network** resulted in a test accuracy of 0.88.

References

1. http://citeseer.ist.psu.edu/517270.html
2. https://pythonhosted.org/xcs/
3. Discussed with classmates.