Assignment MOD 323

To pass the assignment, you shall complete at least one of these exercises.

The three exercises are of increasing complexity, computationally and programming wise. Each one explores one of the problem families of Machine learning.Of course, for a better learning outcome, I would suggest you to complete them all :)

i) Reinforcement learning

Based on the Jupyter Notebook (grid example) discuss (and answer) following questions:

a) Do sensitivity analysis on effect of Number of Episodes on the Reward of each Episodes (change episodes in the code)

- Plot the Episode Reward Vs Episode Number, for episode number from 10 to 100, what do you observe?

b) Change the epsilon value in the code to see tradeoff between “Exploration and Exploitation

- Plot the Total Reward (sum of reward for all episodes) Vs epsilon value (from 0.1 to 0.9). What do you observe in the plot?

- Discuss results of the plot and make discussion about “Exploration VS Explotaion” in RL algorithm

ii) Unsupervised learning

In this exercise you will compare NN vs Linear regression

- Generate a distribution made out of 4 different distributions (use the class sum to generate the data)

- Train a RL and a NN model

- Test the RL and the NN models

- Make a new distribution of data and Validate the RL and NN models on it.

- Compute the residual (sum of square error) for your train, **test** and validation dataset.

- Plot the **test** residual versus the number of total nodes of the NN

- For the best residual found, compute the computational time used by LR and NN.

iii) Unsupervised learning

This exercise will show you how much clustering outcome is sensitive to noise.

Make a data generation class that:

- reads the file ‘xeek_train_subset_mini.csv’

- add noise around ‘RHOB’ and ‘GR’, normally distributed

- visualise the plot with and without noise

- make a k-means and a GMM, and plot the results as a function of DEPTH.

- make a cross-plot with pairplot

- define a sufficient categorization accuracy

- Determine the level of noise that allows you to have still a good categorization (with a python script even better)