Project 2

Part I:

1.

i.

The data set is not ready to be used by the Orange3 neural network as it has many missing values in variables *(HSGPA = 448, LAST_ACT_ENGL_SCORE = 1001, LAST_ACT_MATH_SCORE = 1004, LAST_ACT_READ_SCORE = 1111, LAST_ACT_SCIRE_SCORE = 1114, LAST_ACT_COMP_SCORE = 1001).* Also, variable 'SEX' needs to be cleaned as it contains unwanted numeric and character values ('?').

ii. I have filtered the missing values using *isna()* function and then replaced the missing values with mean of their respective variables. I have chosen to not drop variables because dropping the variables would cause a significant loss of data. Also, the aim of this task is to attract the students with relatively high entrance qualifications. So we should retain the last scores in ACT for different subjects. I have not chosen to replace missing values by zero as it can induce new bias.

For SEX variable I have defined a function to transform the SEX variables from 'M' to 0,' F' to 1, '?' To 2, and numeric values to 3. This simplifies process of dropping rows with'?' and numeric values by filtering 2 and 3 values by index and applying *drop()* function.

Also, I transformed *At_Risk* variable from {F,T} to {0,1}. It makes the data consistent to work with and is always good to convert all character values to numeric.

iii. Python code Attached in zip file.

2.

I have chosen following values for parameters:
Number of hidden layers: 2
Number of neurons: 10(in each layer)
Maximum number of iterations: 2000

Experiment:

- I tried changing the Number of hidden layers to 1, 3, 4 but it gavebest results in 2 layers.
- I also changed the Number of neurons from 10 to 20 but it decreased the accuracy of the algorithm.
- Increasing and decreasing iteration had little effect on accuracy but increased the processing time.

i. Python code Attached in zip file.
ii. I got Accuracy: 0.994 and AUC: 0.999
iii. As in the data set input variables are closely correlated with the outcome, we should use NN classifier. Also, we want the University to select and attract a student body with

relatively high entrance qualifications, so accuracy matters the most. Therefore, NN classifier is best for this problem.

3.

i. Python code Attached in zip file.

ii. Python code Attached in zip file.

iii. As in given the output of python code for confusion matrix we can see that MLP classifier is able to predict 463 labels (diagonally adding true positive (410) and true negative values (53) in the confusion matrix generated) correctly out of 467, which is pretty good prediction rate. The prediction results show that algorithm can predict the labels with 0.99% accuracy.

Note: Accuracy can vary slightly as data is shuffled before splitting into test and training data which changes the model's training process.

Part II.

a. Python code Attached in zip file.

b. Python code Attached in zip file.

Output is:

	y="g"	y="b"
Cluster 0	68	93
Cluster 1	157	33