MATH 6315 - Internship Technical Report

Working as Graduate Student Researcher for HPE-DSI at UH

By: Thomas Zhenkang Su

PHID#: 1616244

Email: zsu8@uh.edu

Abstract

We are all aware of how deep learning has revolutionized our world in recent years and has made a variety of complex tasks much easier to perform. The recent breakthroughs in implementing deep learning techniques has shown that superior algorithms and complex architectures can impart human-like abilities to machines for specific tasks. In the project of my summer internship, it's aimed to assess and summarize the overall predictive ability of multi-layer perceptron (MLP), a subset of deep neural networks, in the topic of Juvenile Delinquents.

Table of Contents

Introduction

My internship was sponsored by the Hewlett Packard Enterprise Data Science Institute (HPE-DSI). The HPE-DSI was launched in 2017 to support and strengthen the university's research and academic interests in data science. The institute puts UH researchers in front of important global challenges by actively promoting community, industry, and institutional partnerships.

I didn't directly work in HPE-DSI, but they appointed me as a Graduate Student Researcher to Genetic and Neurobehavioral Systems: Interdisciplinary Studies (GENESIS) Lab in the Department of Psychology at the University of Houston. My daily task was mainly to assist lab's experienced researchers in handling, cleaning, and analyzing large datasets for their big projects with multiple topics using programming tools Python and R.

In this technical report, one of the projects worked in the GENESIS lab, on the topic of Juvenile Delinquents. In the project of Juvenile Delinquents, my fellow researchers aimed to establish a source of expertise to elucidate the etiology of the empirical overlap between severe Learning Disability (LD) and juvenile delinquency. More specifically, It's to implement a number of activities, as outlined in the ADMINISTRATION CORE, that will crystallize relevant expertise on the overlap between severe LD and juvenile delinquency, establish a network of scientists and practitioners relevant to these issues. As a future data scientist, I was responsible for utilizing the Neural Network Model as a series of algorithms to test features that predict whether the juvenile subject who committed the offense will re-offend before he/she turns 17. What is a Neural Network Model and why did I choose it? Neural Network is a technique for building a computer program that learns from data, and it's based very loosely on how we think the human brain works. First, a collection of software "neurons" are created and connected

together, allowing them to send messages to each other. Next, the network is asked to solve a problem, which it attempts to do over and over, each time strengthening the connections that lead to success and diminishing those that lead to failure.

Analytical Dataset Summary

The analytical dataset is extracted from three different raw datasets named **Offense**, **Detention**, **MAYSI**, from the P-20 longitudinal statewide database. It includes 83498 row observations and 22 column attributes (21 predictor attributes and 1 target attribute).

```
print(df_a.drop(['PID'], axis = 1))
```

```
      JDOF_reoffense  IIBY___  IIBM___  IIBD___  ...  JDMYSC  JDMYSI  JDMYTD  JDMYTE
0               Nope     1990        3        1  ...       2       0       0       0
1               Nope     1990        9       12  ...       1       0       0       0
2                Yep     1991        3        4  ...       4       4       0       1
3               Nope     1991        3        4  ...       0       0       0       0
4                Yep     1991       10        1  ...       3       1       0       3
...              ...      ...      ...      ...  ...     ...     ...     ...     ...
83493           Nope     2008        8        8  ...       1       0       1       3
83494           Nope     2003       10       16  ...       4       1       1       1
83495           Nope     2003        5       28  ...       5       2       1       5
83496           Nope     2005        1       23  ...       3       5       2       2
83497           Nope     2006        8       15  ...       0       0       0       0

[83498 rows x 22 columns]
```

01. **JDOF_reoffense** (String)**:** This feature column indicates the value of the target attribute that if the juvenile subject re-offend before he/she turns 17 as the minimum adult age that the State of Texas recognizes. The string value in this column is either "**Yep**" or "**Nope**".

02 - 04. **IIBY___, IIBM___, IIBD___** (Integer)**:** These three feature columns are, respectively, birth year, birth month, and birth day of the juvenile subject.

05. **IIGE___** (Integer)**:** The gender of the Juvenile Subject. 1 is male and 2 is female.

06. **IIRA___** (Integer)**:** The ethnicity of the Juvenile Subject. 1 is Asian, 2 is Black, 3 is Hispanic, and 4 is White.

07. **JDOF_age** (Numeric)**:** The age of the juvenile subject when he/she committed the offense. The age range should be between 10 and 17.

08. **JDOFND** (Integer)**:** New offense code categorized by severity - Description. It contains 611 unduplicated integer values from 1 to 611, such as, 1, 2, 3, 4, 5, and etc, which corresponds to its original categorical values in alphabetical order, "**A POSS CS PG3<28G DRUG FREE ZO**", "**A TRAFFICKING PRSN PROH COND F**" "**A UNLAW USE OF CRIMINAL INSTR**", "**ABANDON-ENDANGER CHILD", "ABANDON-ENDANGER CHILD-NO RET.**", "**ABANDON-ENDANGER CHILD/RETURN**", and etc..

09. **JDOFCT** (Integer)**:** New offense code categorized by severity - Category. It contains 9 unduplicated integer values from 1 to 9, which corresponds to its original categorical values "**ADMINISTRATIVE**", "**ALL OTHERS/F MAB**", "**CHINS**", "**ILLEGAL SUBSTANCE/F M**", "**PERSON/FELONY**", "**PERSON/MISD**", "**PROPERTY/FELONY**", "**PROPERTY/MISD**", "**VIOLATION OF PROBATION**".

10. **JDOFSL** (Integer)**:** New offense code categorized by severity - Degree of severity, that is in the range from 0 (lowest) to 9 (highest).

11. **JDOF_ordinal** (Integer)**:** Ordinal number that the juvenile subject committed the offense.

12. **JDDT_convict** (Boolean)**:** Is the juvenile subject imprisoned? 1 is true, 0 is false.

13. **JDDT_duration** (Integer)**:** The length of the juvenile subject sentence. The unit is day.

14. **JDMYA** (Boolean)**:** Was the MAYSI-2 administered to the juvenile subject? 1 means true, 0 means false.

15. **JDMY_age** (Numeric)**:** The age of the juvenile subject when MAYSI-2 is administered to him/her.

16 - 22. **JDMYAD**, **JDMYAI**, **JDMYDA**, **JDMYSC**, **JDMYSI**, **JDMYTD**, **JDMYTE** (Integer)**:** These seven feature columns are scores on six clinical scales and one additional scale from **MAYSI** raw dataset.

The MAYSI-2 is a brief behavioral health screening tool designed especially for juvenile justice programs and facilities. It identifies youths 12 through 17 years old who may have important, pressing behavioral health needs. Its primary use is in juvenile probation, diversion programs, and intake in juvenile detention or corrections. The MAYSI-2 is a self-report inventory of 52 questions that ask the youth to answer YES or NO to having experienced various thoughts, feelings or behaviors in the past few months. Answers provide scores on 7 scales: Alcohol/drug use (**JDMYAD**), Angry-irritable (**JDMYAI**), Depressed-anxiety (**JDMYDA),** Somatic complaints (**JDMYSC**), Suicide ideation (**JDMYSI**), Thought disturbance boys (**JDMYTD**), and Traumatic experiences (**JDMYTE**). Alcohol/Drug Use identifies youths who may be using alcohol or drugs to a significant degree, and who are therefore at risk of substance dependence and/or abuse. Angry-Irritable assesses explicit feelings of preoccupying anger, a general tendency toward irritability, frustration, impulsive reactions, and tension related to anger. Depressed-Anxious elicits symptoms of mixed depressed mood and anxiety. Somatic Complaints assess the presence of various bodily aches and pains that may affect the youth, especially those often associated with anxiety. Suicide Ideation specifically identifies thoughts and intentions about self-harm and depressive symptoms that may present an increased risk for suicide. Thought Disturbance *(boys only)* assesses the possibility of serious mental disorder involving distortion of reality. (In the study with which the MAYSI-2 was developed and validated, a scale

associated with thought disturbance proved satisfactory for use with boys, but did not prove satisfactory for use with girls. Its meaning when administered to girls, therefore, is questionable). The additional seventh scale is Traumatic Experiences. While this scale does not assess a clinical condition, like PTSD, it identifies youths who may have had greater exposure to potentially traumatizing events compared to other youths, not whether they were traumatized. However, each scale represents some important set of behaviors, thoughts, feelings or symptoms associated with various behavioral health problems of adolescents. These MAYSI assessments' scores are in the range from 0 (lowest) to 9 (highest).

```
> summary(df_a)
    IIBY___        IIBM___          IIBD___          IIGE___
 Min.   :1988   Min.   : 1.000   Min.   : 1.00   Min.   :1.00
 1st Qu.:1994   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:1.00
 Median :1996   Median : 7.000   Median :16.00   Median :1.00
 Mean   :1997   Mean   : 6.602   Mean   :15.63   Mean   :1.23
 3rd Qu.:2000   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1.00
 Max.   :2008   Max.   :12.000   Max.   :31.00   Max.   :2.00
    IIRA___         JDOF_age         JDOFND          JDOFCT
 Min.   :1.000   Min.   :10.01   Min.   :  1.0   Min.   :1.00
 1st Qu.:2.000   1st Qu.:14.52   1st Qu.: 92.0   1st Qu.:4.00
 Median :3.000   Median :15.54   Median :335.0   Median :6.00
 Mean   :2.715   Mean   :15.29   Mean   :308.8   Mean   :5.45
 3rd Qu.:3.000   3rd Qu.:16.31   3rd Qu.:522.0   3rd Qu.:8.00
 Max.   :4.000   Max.   :17.00   Max.   :611.0   Max.   :9.00
     JDOFSL        JDOF_ordinal     JDDT_convict    JDDT_duration
 Min.   :0.000   Min.   : 1.000   Min.   :0.0000   Min.   :  0.000
 1st Qu.:4.000   1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.:  0.000
 Median :5.000   Median : 1.000   Median :0.0000   Median :  0.000
 Mean   :4.696   Mean   : 1.883   Mean   :0.4679   Mean   :  6.524
 3rd Qu.:6.000   3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:  2.000
 Max.   :9.000   Max.   :15.000   Max.   :1.0000   Max.   :673.000
    JDMY_age         JDMYA            JDMYAD           JDMYAI
 Min.   :10.02   Min.   :0.0000   Min.   :-1.0000   Min.   :-1.000
 1st Qu.:14.62   1st Qu.:0.0000   1st Qu.:-1.0000   1st Qu.:-1.000
 Median :15.64   Median :1.0000   Median : 0.0000   Median : 0.000
 Mean   :15.39   Mean   :0.6053   Mean   : 0.4698   Mean   : 1.828
 3rd Qu.:16.41   3rd Qu.:1.0000   3rd Qu.: 1.0000   3rd Qu.: 4.000
 Max.   :17.48   Max.   :1.0000   Max.   : 8.0000   Max.   : 9.000
     JDMYDA           JDMYSC           JDMYSI
 Min.   :-1.0000   Min.   :-1.0000   Min.   :-1.00000
 1st Qu.:-1.0000   1st Qu.:-1.0000   1st Qu.:-1.00000
 Median : 0.0000   Median : 0.0000   Median : 0.00000
 Mean   : 0.8992   Mean   : 0.9863   Mean   : 0.03625
 3rd Qu.: 2.0000   3rd Qu.: 3.0000   3rd Qu.: 0.00000
 Max.   : 9.0000   Max.   : 6.0000   Max.   : 5.00000
     JDMYTD           JDMYTE
 Min.   :-1.0000   Min.   :-1.0000
 1st Qu.:-1.0000   1st Qu.:-1.0000
 Median : 0.0000   Median : 0.0000
 Mean   : 0.0114   Mean   : 0.4522
 3rd Qu.: 0.0000   3rd Qu.: 1.0000
 Max.   : 6.0000   Max.   : 5.0000
```

Brief Introduction to MLP Model

Multilayer Perceptron (MLP) is a type of feedforward Artificial Neural Network Model that introduces one or more hidden layers on the basis of a single-layer neural network. MLP is mainly used for pattern classification, recognition, prediction and approximation. An arbitrary number of hidden layers that are placed in between the input layer and the output layer. The input layer doesn't involve any calculation, but the output of each hidden layer is transformed by an activation function, such as ReLU (rectified linear unit) function, sigmoid function, and hyperbolic tangent (tanh) function. Without an activation function applied to each hidden layer and an output layer, MLP functions as a single-layer artificial neural network that the connected layer performs the affine transformation (that is any transformation that preserves collinearity), and the superposition of multiple affine transformations is still the same as an affine transformation. The solution is to introduce the nonlinear transformation, well known as the activation function.

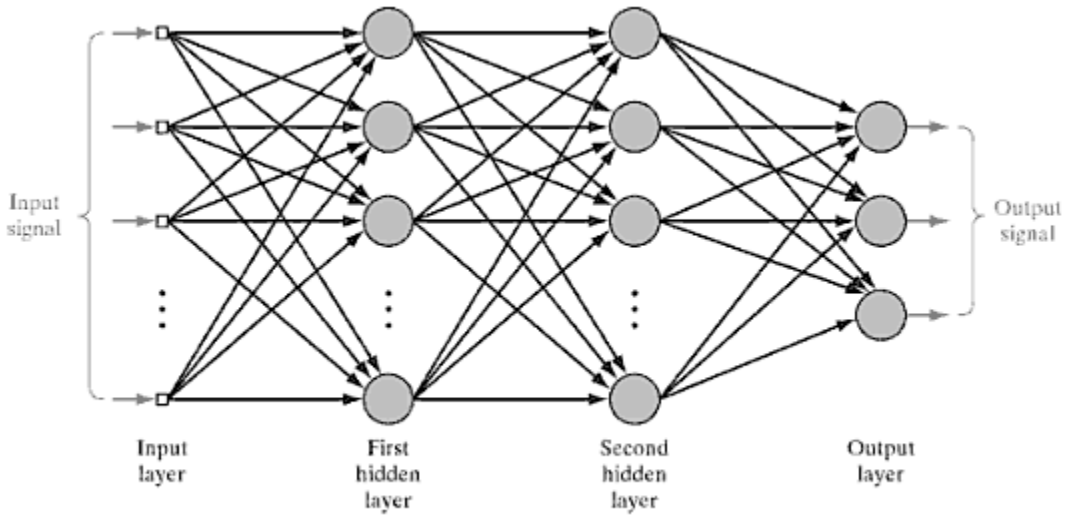


FIGURE 4.1 Architectural graph of a multilayer perceptron with two hidden layers.

MLP Model Set-up

Before launching the training of the MLP model, we have to partition the analytical dataset into 70% for the training set and 30% for the validation set, because the MLP model utilizes supervised deep learning technique that learns the past input data and makes future predictions as output. Also, the analytical dataset should be standardized for speeding up learning and leading to faster convergence. After that, we need to configure it as determining the categorical cross entropy function, activation function, an optimal amount of neurons in the hidden layer, an optimal number of the hidden layer, the learning rate, the batch size, and etc..

The categorical cross entropy is used as a loss function. Because there are two or more label classes, and labels are provided in a one-hot array containing the probable match for each class (category).

The choice of the activation function is dependent on the use case and the data. But for simpler datasets, the activation function will not make a huge difference. The neuron activation functions for two hidden layers are randomly chosen as, respectively, ReLu, sigmoid. The softmax activation function is applied to the output layer.

Generally, for an MLP model, making a model extremely complex by adding a lot of hidden layers is never a good idea as it makes the model more complex, slow and can lead to overfitting as well. Therefore, two or fewer layers will often suffice with simple data sets. With complex datasets involving time-series or computer vision, additional layers can be helpful. In the model, the number of the hidden layers is determined as 2, and it's believed the result can

represent an arbitrary decision boundary to arbitrary accuracy with rational activation functions and can approximate any smooth mapping to any accuracy.

As for determining an acceptable number of neurons to use in the hidden layers, there are many rule-of-thumb approaches, such as the following: the number of hidden neurons should be between the size of the input layer and the size of the output layer, the number of hidden neurons should be the sum of 2/3 the size of the input layer and the size of the output layer, and the number of hidden neurons should be less than twice the size of the input layer. However, in the model, the first two approaches are applied. The neuron amounts of the first and second hidden layer are, respectively, 40 and 16.

The learning rate is an important factor as it is used to find the best solution for the model. A model makes mistakes and learns from it and the learning rate controls how quickly or slowly a neural network model learns a problem. If the learning rate is too small, the model will take a long time to reach the best possible stage. But if it is too high, the model might just skip the best stage. In the model, we stick with the default of the learning rate that is set to 0.001.

The batch size is a hyperparameter that we must test and tune based on how the specific model is performing during training. By grouping the data in batches, the quicker the model will complete each epoch during training. This way, we are able to process much more than one single sample at a time. The batch size is followed by the function int(sqrt(x)), where x is the row number of the training set.

The number of epochs is also a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. An epoch consists of one or more batches. To determine the epoch size for the MLP model, it’s

common to create line plots that show epochs along the x-axis as time and the error or skill of the model on the y-axis. These plots are sometimes called learning curves. These plots can help to diagnose whether the model has over learned, under learned, or is suitably fit to the training dataset.

To reduce any possible effects of overfitting, an error that can happen when a network is too closely fit to the input samples by ignoring a random subset of units in a layer while setting their weights to zero during the phase of training, the MLP model is added a simple but powerful regularization technique called Dropout to all input layers, hidden layers and output layers to drop some nodes of the network in order to temporarily deactivate or ignore neurons. However, the dropout rate for the hidden layers is set as 0.4, whereas the dropout rate for the output layers is set as 0.2.

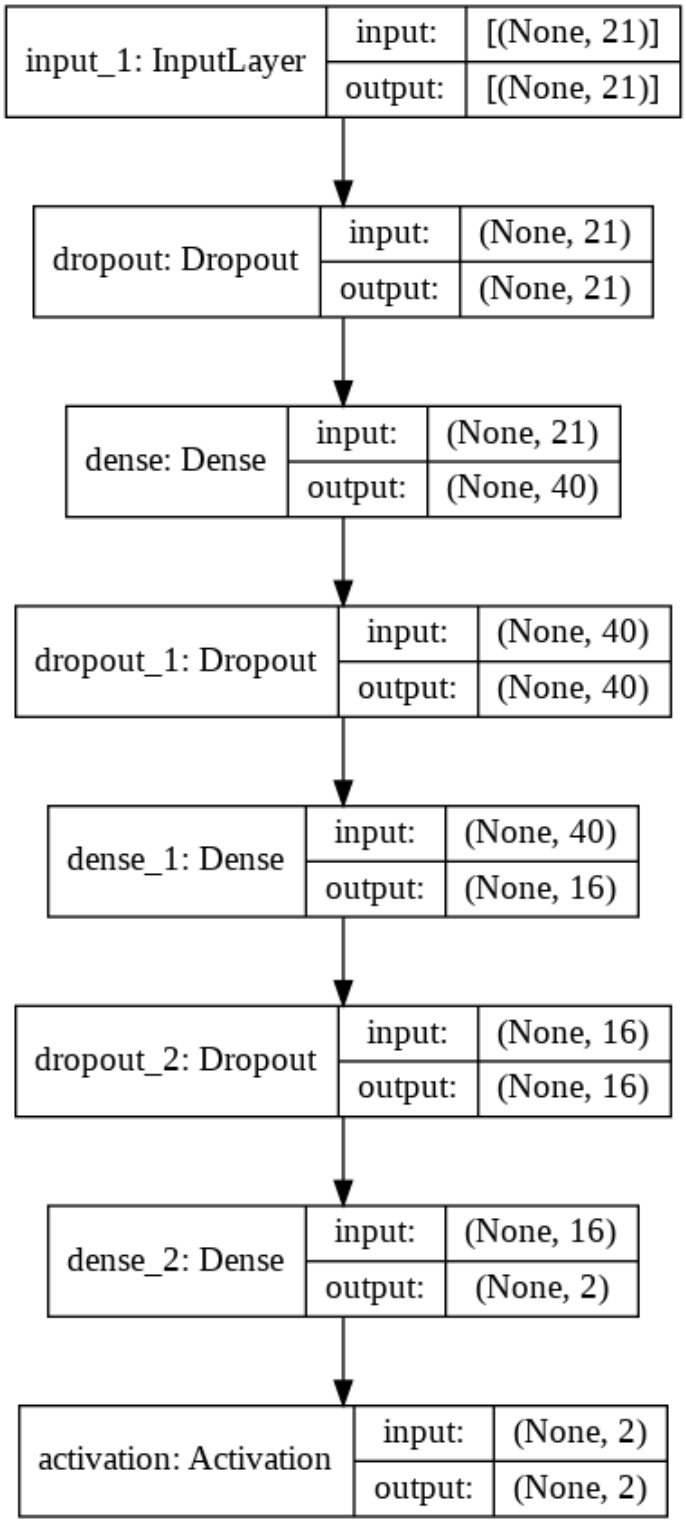
Customized MLP Model Architecture

Input → Dropout 1 → Hidden 1 → Dropout 2 → Hidden 2 → Dropout 3 → Output By Softmax → Probability

```
trained_model_1.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dropout (Dropout)            (None, 21)                0
_________________________________________________________________
dense (Dense)                (None, 40)                880
_________________________________________________________________
dropout_1 (Dropout)          (None, 40)                0
_________________________________________________________________
dense_1 (Dense)              (None, 16)                656
_________________________________________________________________
dropout_2 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 34
_________________________________________________________________
activation (Activation)      (None, 2)                 0
=================================================================
Total params: 1,570
Trainable params: 1,570
Non-trainable params: 0
_________________________________________________________________
```

```
from tensorflow.keras.utils import plot_model
plot_model(model = trained_model_1, show_shapes = True, show_layer_names = True)
```



Model Training Performance

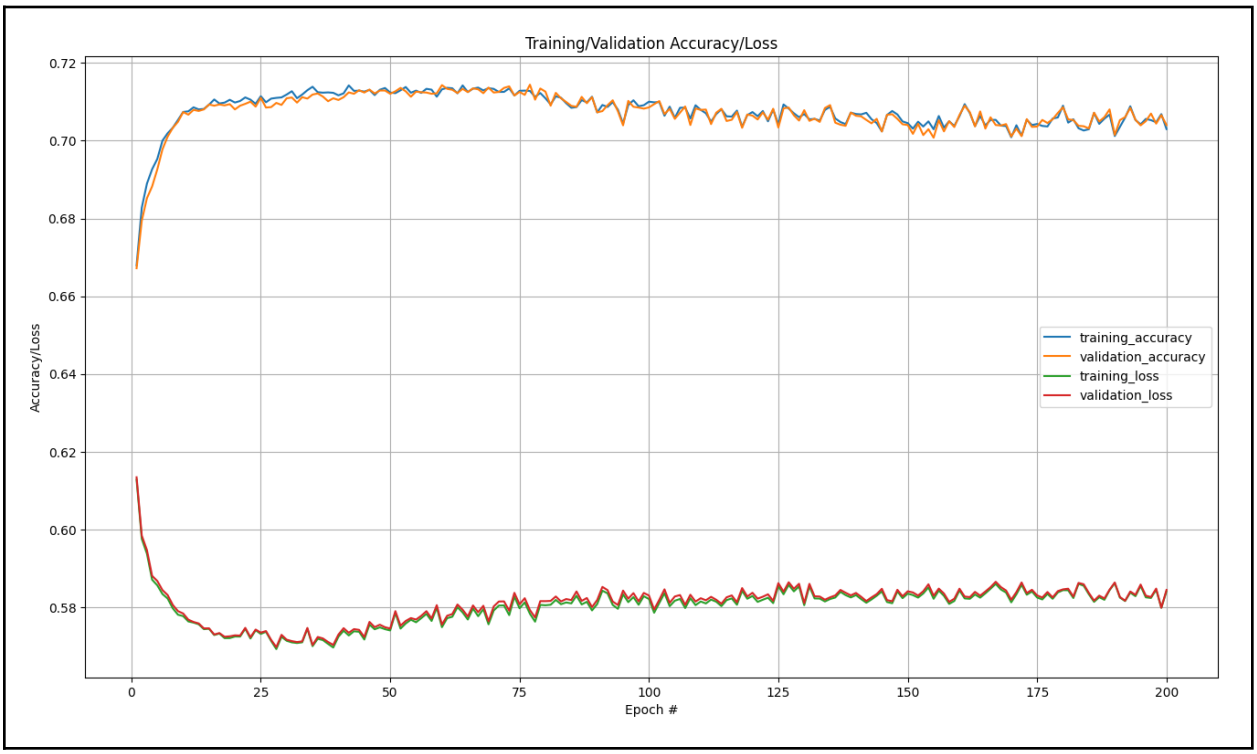
After the MLP model is trained, the next step is to diagnose its learning performance by observing learning curves that show the number of epochs on the x-axis and accuracy or loss on the y-axis. Observing the graph of Training/Validation Loss, A model is neither underfitting that will have high training and low validation loss nor overfitting that will have extremely low training error but a high validation loss. But, the training loss and the validation loss moved close to each other at all times with validation loss being slightly greater than the training loss. Also, training loss and validation loss is initially decreasing and training and validation loss after some point till the end turn out to be flat. It reflects the model performed very well, and it's identified as a good fit model. However, no extra adjustments to the model need to be made.

As for the graph of Training/Validation Accuracy, it can be discovered that the best accuracy exists between epoch 25 and epoch 75. After epoch 75, the accuracy started decreasing to a point of stability. So, the accuracy of the MLP model won't keep increasing if the model trains forever, and the number of epochs can be set as below 100 to avoid consuming extra time on the model training.

Based on the monitor of the model displayed, the best performance of the training set is achieved as 71.28%, which corresponds to 0.5783 as its training loss. The best performance of the validation set is achieved as 71.44%, which corresponds to 0.5792 as its validation loss.

As we see confusion matrices for both training set and validation set, the darker diagonal values are the percentage of correct classifications for the classes, and everything else (lighter blue) is the percentage of times the model was not correct in predicting that class. Overall, our confusion matrix reports that our model performed without much error at all with label

prediction accuracies of 59% and 80% for our two classes. The confusion matrix for the validation set reported moderately high prediction accuracies of 58% and 81% for classes “Yep” and “Nope”.



```
best_epoch_1 = monitor_1.iloc[monitor_1['val_accuracy'].idxmax(), ]
```

```
train_perf_1 = best_epoch_1['train_accuracy']
train_perf_1
```

```
0.7128120660781859
```

```
test_perf_1 = best_epoch_1['val_accuracy']
test_perf_1
```

```
0.7144277691841125
```

```
train_loss_1 = best_epoch_1['train_loss']
train_loss_1
```

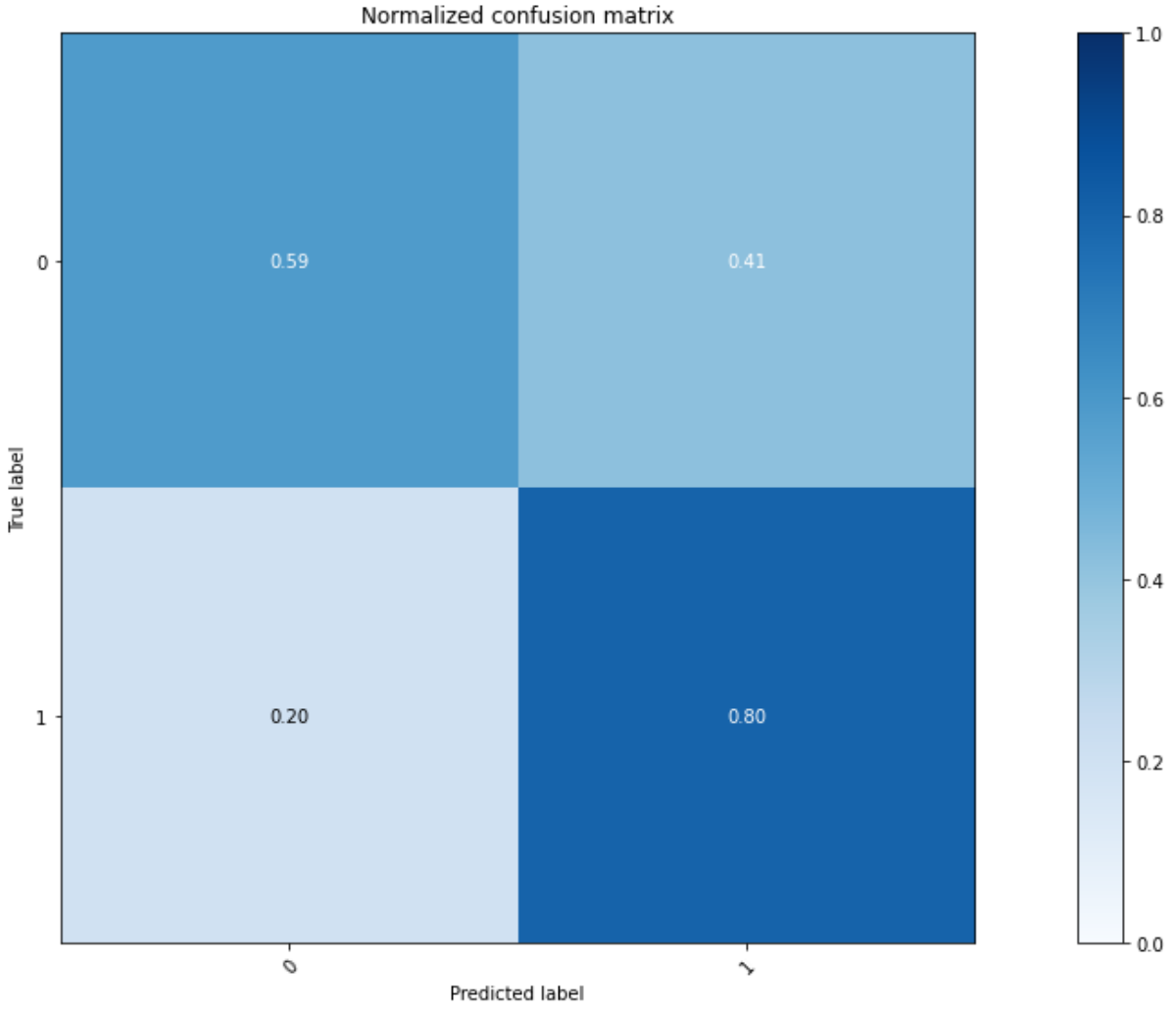
```
0.5783394575119019
```

```
test_loss_1 = best_epoch_1['val_loss']
test_loss_1
```

```
0.5792912244796753
```

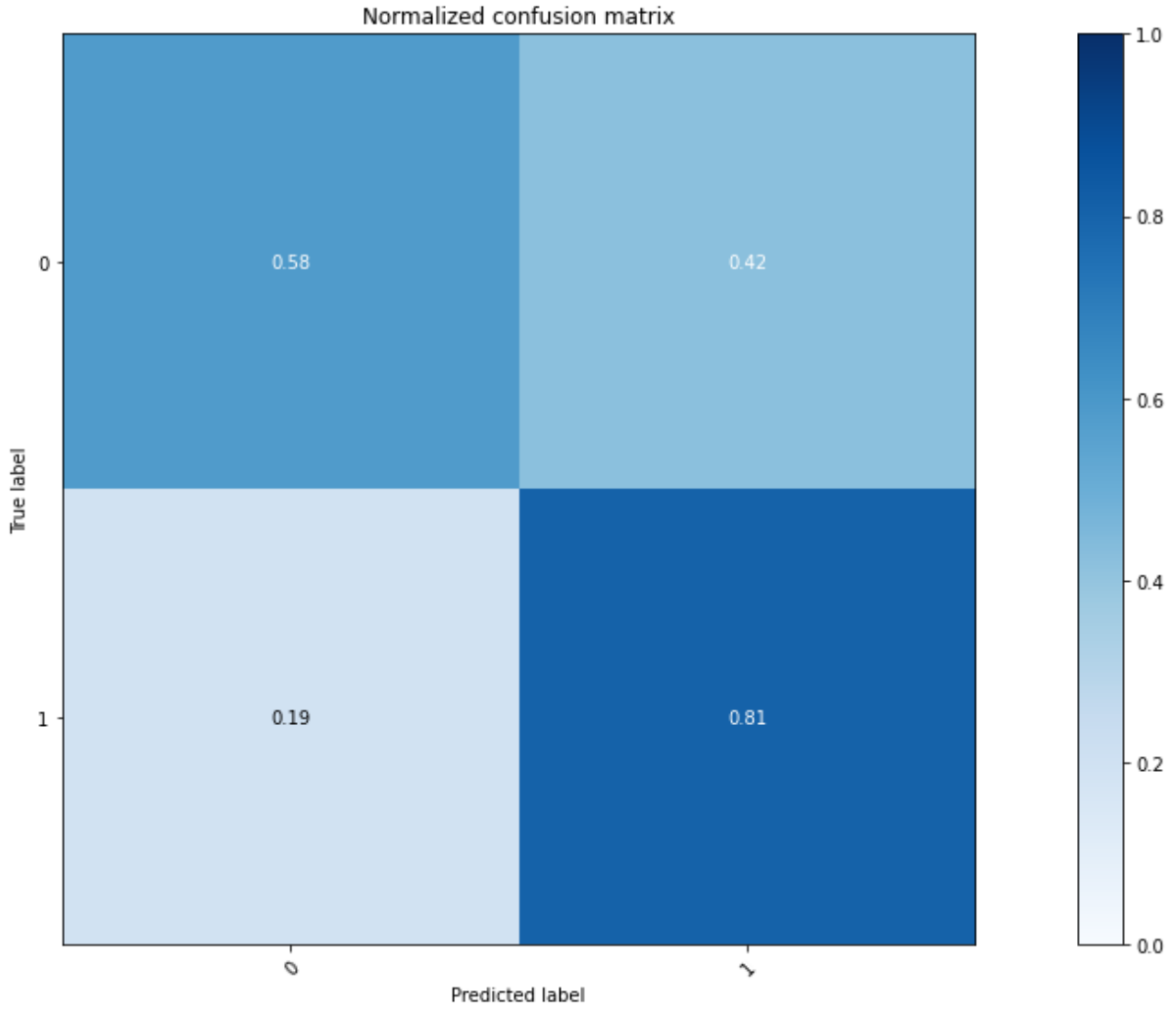
Confusion Matrix for Training Set:

```
Normalized confusion matrix
[[0.58502443 0.41497557]
 [0.19700114 0.80299886]]
```



Confusion Matrix for Validation Set:

```
Normalized confusion matrix
[[0.58076102 0.41923898]
 [0.19276134 0.80723866]]
```



Prediction Attempt

Prediction ability is an ultimate goal in this report. After the MLP trained and its performance analyzed while training, the model can be used to predict. Assume that, there was a case that male Asian juvenile subject who was borned on January 1, 2001 committed his first offense on February 6, 2016, and he was charged with MURDER that is categorized by degree 9. He was convicted of murder and imprisoned from February 7, 2016 to December 1, 2018. He took a MAYSI-2 assessment on March 4, 2016. The Alcohol/drug use (**JDMYAD**) score, Angry-irritable (**JDMYAI**) score, the Depressed-anxiety (**JDMYDA**) score, the Somatic complaints (**JDMYSC**) score, the Suicide ideation (**JDMYSI**) score, the Thought disturbance boys (**JDMYTD**) score, and the Traumatic experiences (**JDMYTE**) score, are, respectively, 5, 4, 3, 2, 1, 0, 0.

```
Enter the Birth Date in YYYY-MM-DD format: 2001-01-01
Enter the Gender: Male
Enter the Ethnicity: Asian
Enter the Offense Date in YYYY-MM-DD format: 2016-02-06
Enter an ordinal number that the juvenile object committed the offense: 1
Enter the description that new offense code categorized by severity: MURDER
Enter the degree of severity that new offense code categorized by severity: 9
Is the juvenile object imprisoned? Yes
Enter the date the juvenile object was placed in detention in YYYY-MM-DD format: 2016-02-07
Enter the date the juvenile object was released from detention in YYYY-MM-DD format: 2018-12-1
Enter Screening date or date the department attempted to administer or an assessment date for MAYSI-2 in Y\
Was the MAYSI-2 administered to the juvenile object? Yes
Enter the Alcohol/drug Use Score: 5
Enter the Angry-irritable Score: 4
Enter the Depressed-anxiety Score: 3
Enter the Somatic Complaints Score: 2
Enter the Suicide Ideation Score: 1
Enter the Thought Disturbance Boys Score: 0
Enter the Traumatic Experiences Score: 0
```

```
import numpy as np
import pandas as pd
input = np.array([IIBY___, IIBM___, IIBD___, IIGE___, IIRA___, JDOF_age, JDOFND, JDOFCT,
                  JDOFSL, JDOF_ordinal, JDDT_convict, JDDT_duration,
                  JDMYA, JDMY_age, JDMYAD, JDMYAI, JDMYDA, JDMYSC, JDMYSI, JDMYTD, JDMYTE])
input = input.reshape(1, input.shape[0])
INPUT = pd.DataFrame(data = input,
                     columns = ['IIBY___', 'IIBM___', 'IIBD___', 'IIGE___', 'IIRA___', 'JDOF_age',
                                'JDOFND', 'JDOFCT', 'JDOFSL', 'JDOF_ordinal', 'JDDT_convict',
                                'JDDT_duration', 'JDMYA', 'JDMY_age', 'JDMYAD', 'JDMYAI', 'JDMYDA',
                                'JDMYSC', 'JDMYSI', 'JDMYTD', 'JDMYTE'], index = ['input'])
```

Through model prediction, it is determined in this case that the probability that the juvenile subject will commit an offense again until he is recognized as an adult is 28.6576%, the reverse one is 71.3424%. Hence, it's unlikely he will re-offend in this case.

```
pd.DataFrame(data = INPUT.loc[:, 'IIBY___' : 'IIRA___'],
             columns = ['IIBY___', 'IIBM___', 'IIBD___', 'IIGE___', 'IIRA___'],
             index = ['input'])
```

	IIBY___	IIBM___	IIBD___	IIGE___	IIRA___
input	2001.0	1.0	1.0	1.0	1.0

```
pd.DataFrame(data = INPUT.loc[:, 'JDOF_age' : 'JDDT_duration'],
             columns = ['JDOF_age', 'JDOFND', 'JDOFCT', 'JDOFSL',
                        'JDOF_ordinal', 'JDDT_convict', 'JDDT_duration'],
             index = ['input'])
```

	JDOF_age	JDOFND	JDOFCT	JDOFSL	JDOF_ordinal	JDDT_convict	JDDT_duration
input	15.098563	350.0	5.0	9.0	1.0	1.0	1028.0

```
pd.DataFrame(data = INPUT.loc[:, 'JDMYA' : 'JDMYTE'],
             columns = ['JDMYA', 'JDMY_age', 'JDMYAD', 'JDMYAI',
                        'JDMYDA', 'JDMYSC', 'JDMYSI', 'JDMYTD', 'JDMYTE'],
             index = ['input'])
```

	JDMYA	JDMY_age	JDMYAD	JDMYAI	JDMYDA	JDMYSC	JDMYSI	JDMYTD	JDMYTE
input	1.0	15.172485	5.0	4.0	3.0	2.0	1.0	0.0	0.0

```
os.chdir('/content/drive/MyDrive/math 6315/Datasets/' + 'best_model_20210723-223457/')
from tensorflow.keras.models import load_model
trained_model_1 = load_model(base_folder_1)
predict = pd.DataFrame(
    data = trained_model_1.predict(input),
    columns = ['Yep', 'Nope'],
    index = ['Probability'])
predict
```

	Yep	Nope
Probability	0.286576	0.713424

Conclusion

In this report, I explored a MLP model built up of several layers. All the layers in MLP are dense i.e. feed forward. Overall, my model trained very well, it never goes underfitting nor overfitting at all times. As for the accuracy of the prediction, the confidence percentage of when the model predicts that the juvenile subject will re-offend before he/she becomes adult is quite low, although the one of when the model predicts that the juvenile subject will not re-offend before he/she becomes adult is high, based on the diagonal numbers in normalized confusion matrices of training set and validation set. Therefore, we need to express reservation when the model says “Yep”. All in all, neural networks are not magical tools that can learn everything, it depends on the “quality” of the dataset. For example, when I cleaned the raw datasets, I discovered that, out of 83498 row cases, there are 32956 cases that the MAYSI-2 was not administered to the juvenile subject, which accounted for a relatively high proportion as nearly 40%. Because 40% of the total cases lack very important information, as scores on six clinical scales and one additional scale. It increases the learning difficulty of the MLP model and the possibility that the model made mistakes, and leads to a bad influence as the accuracy of the model is only 71%. To improve my model, I think, next time, I can try a combination of dense layers along with CNN, RNN, LSTM, GRU or other layers.

Reference

https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron

https://www.sciencedirect.com/topics/computer-science/confusion-matrix

https://www.sciencedirect.com/science/article/pii/S0004370214000216

https://www.thetexasattorney.com/juvenile-criminal-defense/#:~:text=The%20state%20of%20Texas%20falls,capacity%20to%20have%20criminal%20intent

http://www.nysap.us/maysi2/index.html