ASSIGNMENT 0

Name: Dharma. Acha
UBIT: dharmaac
UB Number: 50511275

About the DataSet :

The data is related to “New York State Statewide COVID-19 Hospitalizations and Beds”. This dataset describes the facility level of the patients hospitalised, admitted, discharged and fatalities. It also includes information about staffed beds. And it also says hospitalised means patients admitted due to covid and not admitted due to covid. The dataset has both categorical and numerical columns.

The dataset link is:
https://health.data.ny.gov/Health/New-York-State-Statewide-COVID-19-Hospitalizations/jw46-jpb7/about_data

Number of Rows: 239947

Number of Columns: 37

I dropped a few columns which are necessary which have no relation with other columns and formed 20 columns.

Key Statistics:

	Patients Currently Hospitalized	Patients Admitted Due to COVID	Patients Admitted Not Due to COVID	Patients Newly Admitted	Patients Positive After Admission	Patients Discharged	Patients Currently in ICU	Patients Expired	Total Staffed Beds	Total Staffed Beds Currently Available	Total Staffed ICU Beds	Total Staffed ICU Beds Currently Available	Total New Admissions Reported
count	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000	239849.000000
mean	15.166642	2.601637	2.886946	1.576884	0.659069	1.920183	2.860091	0.233509	80.733991	22.451196	10.915997	3.657901	1.909451
std	32.631431	8.145138	8.648461	3.455791	2.725286	3.831080	8.672962	0.945901	176.789744	48.492571	29.034616	10.418419	3.564467
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	5.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
75%	16.000000	2.000000	2.000000	2.000000	0.000000	2.000000	3.000000	0.000000	70.000000	22.000000	7.000000	1.000000	2.000000
max	692.000000	160.000000	177.000000	90.000000	80.000000	90.000000	254.000000	27.000000	1313.000000	538.000000	340.000000	129.000000	55.000000

	Young	Adult	Senior
	239849.000000	239849.000000	239849.000000
	0.251775	5.833312	8.430579
	1.178861	14.982343	17.967295
	0.000000	0.000000	0.000000
	0.000000	0.000000	0.000000
	0.000000	1.000000	2.000000
	0.000000	6.000000	9.000000
	42.000000	365.000000	347.000000

Median of the dataset:

```
Patients Currently Hospitalized              5.0
Patients Admitted Due to COVID               0.0
Patients Admitted Not Due to COVID           0.0
Patients Newly Admitted                      0.0
Patients Positive After Admission            0.0
Patients Discharged                          1.0
Patients Currently in ICU                    0.0
Patients Expired                             0.0
Total Staffed Beds                           0.0
Total Staffed Beds Currently Available       0.0
Total Staffed ICU Beds                       0.0
Total Staffed ICU Beds Currently Available   0.0
Total New Admissions Reported                1.0
Young                                        0.0
Adult                                        1.0
Senior                                       2.0
dtype: float64
```

Mode of the dataset:

Facility Name: HEALTHALLIANCE HOSPITAL BROADWAY CAMPUS
DOH Region : METROPOLITAN AREA REGIONAL OFFICE
Facility County: NEW YORK
Facility Network: INDEPENDENT

Missing Values:

Column	Missing count
DOH Region	2
Facility County	5

Facility Network	63
Patients Currently Hospitalized	1
Patients Admitted Due to COVID	2
Patients Admitted Not Due to COVID	4
Patients Newly Admitted	1
Patients Discharged	2
Patients Currently in ICU	3
Patients Expired	6
Total Staffed Beds	5
Total Staffed Beds Currently Available	5
Total Staffed ICU Beds	2
Adult	1

Above columns have missing values which are less than 5% and so dropped the rows with missing values.

Data Preprocessing:

Handling missing values:

I dropped the columns which have null values, Since there are many rows these dropped rows do not influence the actual dataset.

Handling Mismatched String:

I converted the strings to upper case to avoid case sensitivity issues. For example:

df['Facility Name']=df['DOH Region'].str.upper() as shown:

['WESTERN REGIONAL OFFICE', 'METROPOLITAN AREA REGIONAL OFFICE',
'CENTRAL NEW YORK REGIONAL OFFICE',
'CAPITAL DISTRICT REGIONAL OFFICE',
'METROPOLITAN AREA REGIONAL OFFICE',
'METROPOLITAN AREA REGIONAL OFFICE']

Handling Outliers:

I plotted box plot for the all numerical columns and understood the range of outliers and handled the using conditional statements and replaced them with medians of respective columns for example:

df['Patients Admitted Due to COVID']=np.where((df['Patients Admitted Due to COVID']>160) ,medians[1],df['Patients Admitted Due to COVID'])

Onehotencoding:

I used onehotencoding to convert all categorical columns to binary columns so that model is able to run the data. The categorical columns are:

['Facility Name','Facility County','Facility Network']

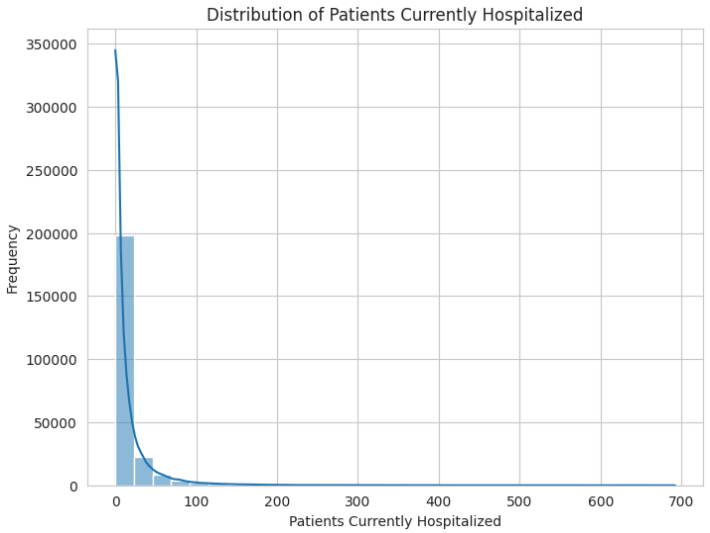
Normalization:

I used the Scikit-Learn library called Standard Scaler for normalization. I normalised all the numerical columns. The numerical columns are:

['Patients Currently Hospitalized','Patients Admitted Due to COVID','Patients Admitted Not Due to COVID','Patients Newly Admitted','Patients Positive After Admission','Patients Discharged','Patients Currently in ICU','Patients Expired','Total Staffed Beds','Total Staffed Beds Currently Available','Total Staffed ICU Beds','Total Staffed ICU Beds Currently Available','Total New Admissions Reported','Young','Adult','Senior']

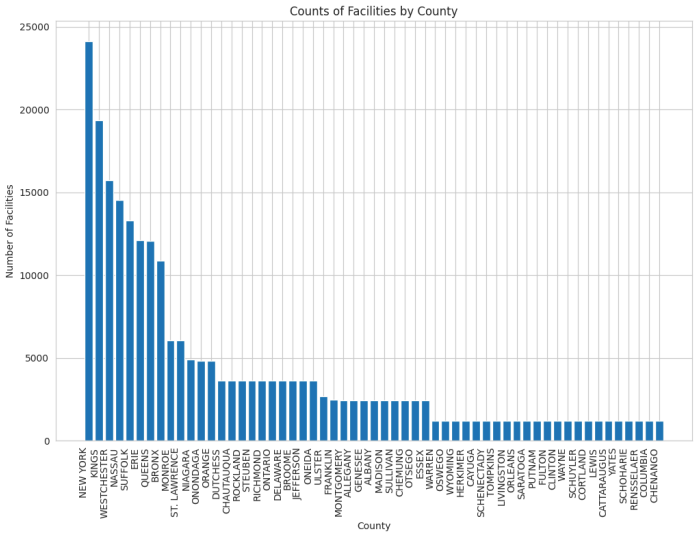
Visualisation Graphs:

Graph 1:



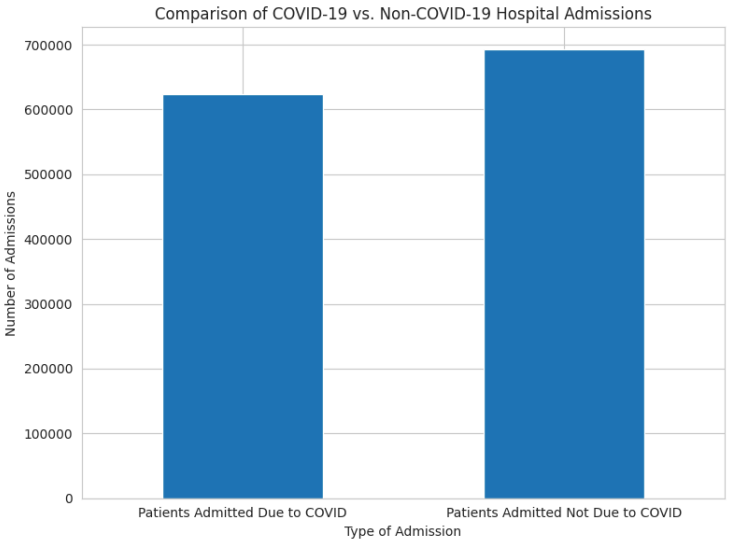
The plot describes the histplot of the patients currently hospitalised. The number of patients currently hospitalised has decreased rapidly.

Graph 2:



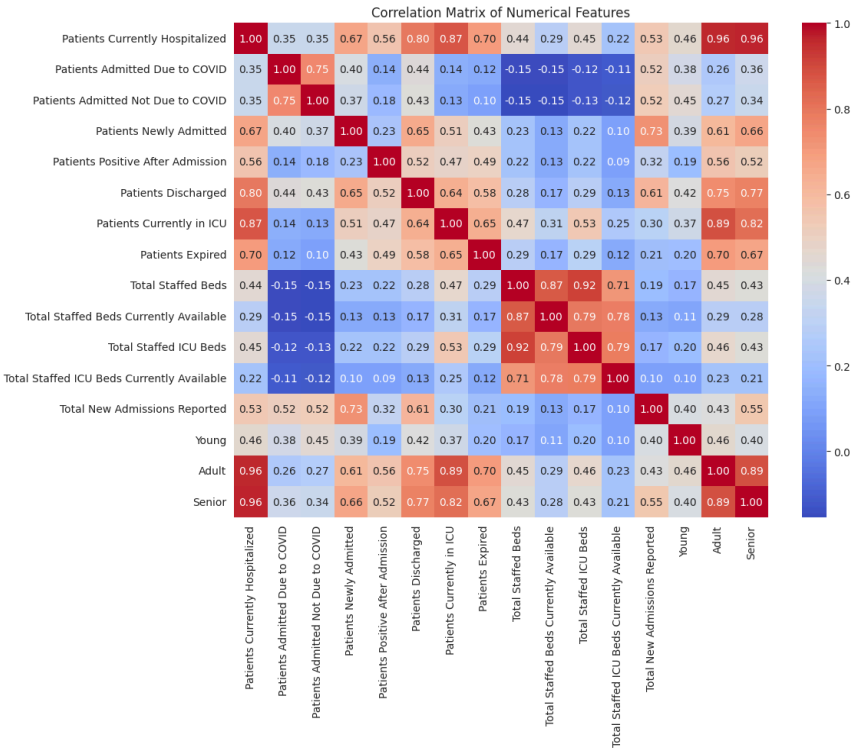
The graph describes the count of the facilities by each county. From the graph we can infer that counties such as New York, Kings, Westchester stood top among all the counties and Chenango, Columbia, yates stood at the bottom of the count.

Graph 3:



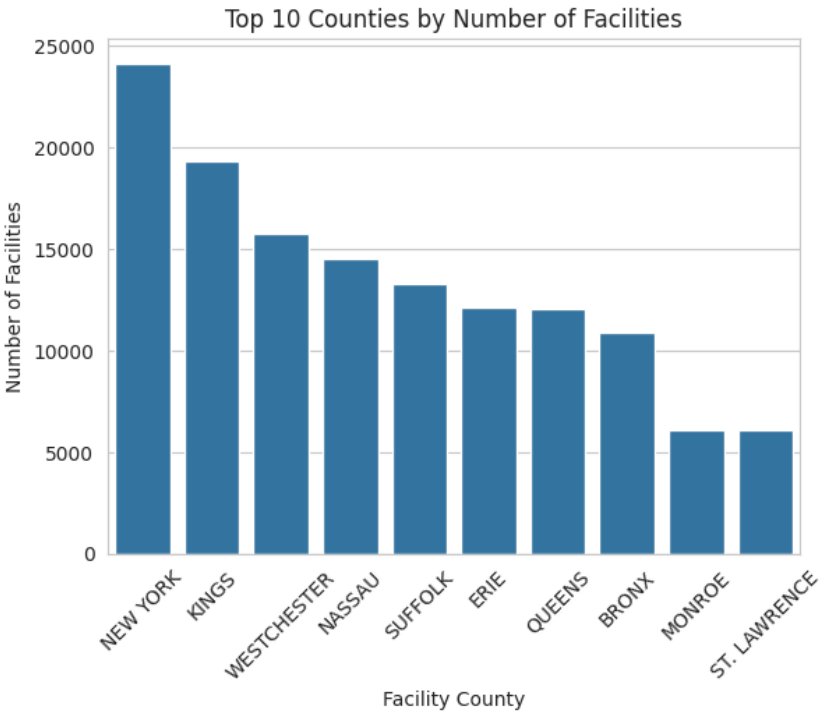
The plot is about comparison of covid and non-covid hospital admissions. From the graph we can say that patients admitted due to covid are less than that of patients admitted not due to covid. There difference is approximately equal to 100000.

Graph 4:



The confusion matrix represents the relation between the columns from the plot. We can say that patients admitted due to COVID and patients not admitted due to COVID have strong relations and it is a positive relation. Whereas between the columns total staffed beds and patients admitted due to COVID has negative correlation likewise we can see correlation among the columns in correlation matrix.

Graph 5:



The graph represents the top 10 counties which have a number of facilities. From the plot we see that New York has the highest number of facilities which approximately equals to 24000 and least one is ST. Lawrence and Monroe which settled at 6000 facilities.

Machine Learning Models:

Models Applied:

Random Forest
Decision Tree
XGBoost

Random Forest: It is used for classification, regression and other tasks. It is ensemble learning, operated by constructing multiple decision trees during training time and output mode or mean of individual trees.

Key Features: Feature randomness, averaging of trees

Advantages:
Handles non linear data well
High Accuracy
Effective in high dimensional spaces

Disadvantages:
Requires high computational resources
Less interpretable

Decision Tree: A decision tree is a flowchart-like tree structure where an internal node represents a feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. It is used for classification and regression tasks.

Key features: Hierarchical, greedy splitting, uses information gain

Mathematical Equation:

Entropy:

$$H(T) = I_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i log_2 p_i$$

Gini Impurity:

$$H(T) = I_G(p_1, p_2, \ldots, p_J) = 1 - \sum_{i=1}^{J} p_i^2$$

Information Gain:

$$IG(T,a) = Entropy(T) - \sum_{a} p(a) Entropy(T \mid a)$$

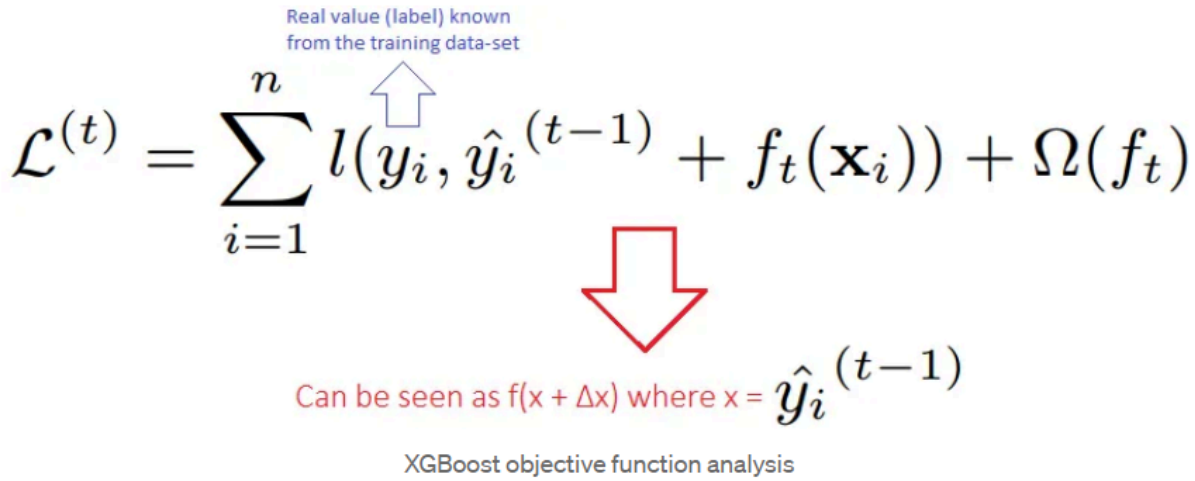
Advantages:
Simple to understand and interpret.
Requires little data preparation.
Can handle both numerical and categorical data.

Disadvantages:
Prone to overfitting
Can be unstable due to high variance

XGBoost: It is an optimised version of the gradient boosting framework. It is used for supervised learning problems, where the goal is to use the training data to predict an outcome given new input data.

Key Features: System optimizations, algorithm enhancements

Mathematical Equation:



XGBoost objective function analysis

Advantages:
Fast and scalable
Highly accurate
Built-in regularization to prevent overfitting.

Disadvantages:
Overfitting is possible if tuning is not done properly
More complex to understand and interpret

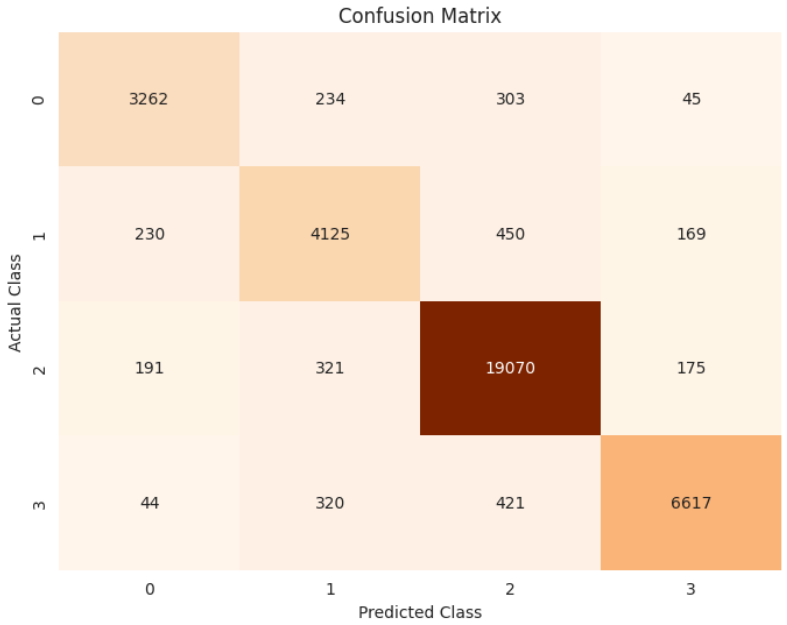
Accuracy and Loss:

Model	Accuracy	Loss
Decision Tree	91.93%	1.39
Random Forest	69.32%	1.39

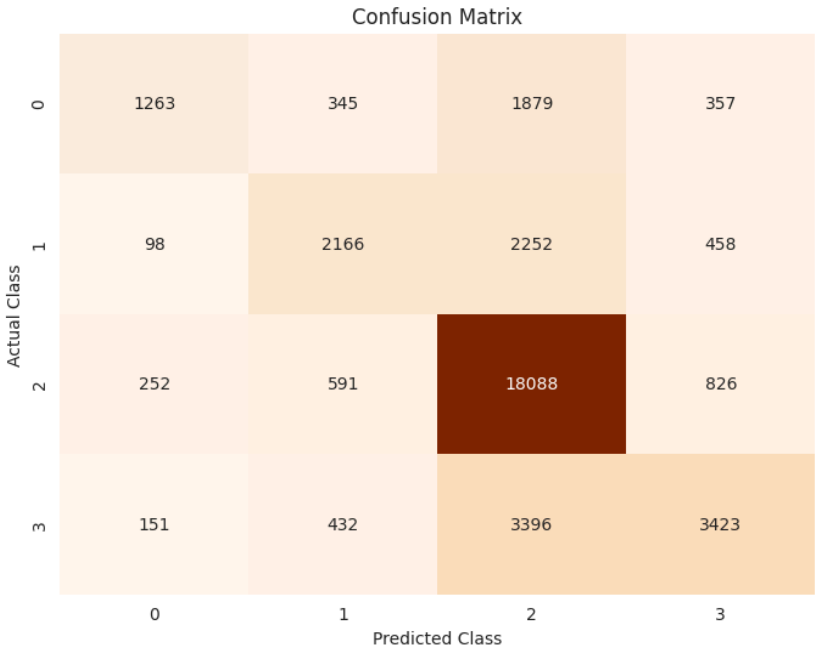
XGBoost	70.52%	0.74

Below is the confusion Matrix of each prediction of the model.

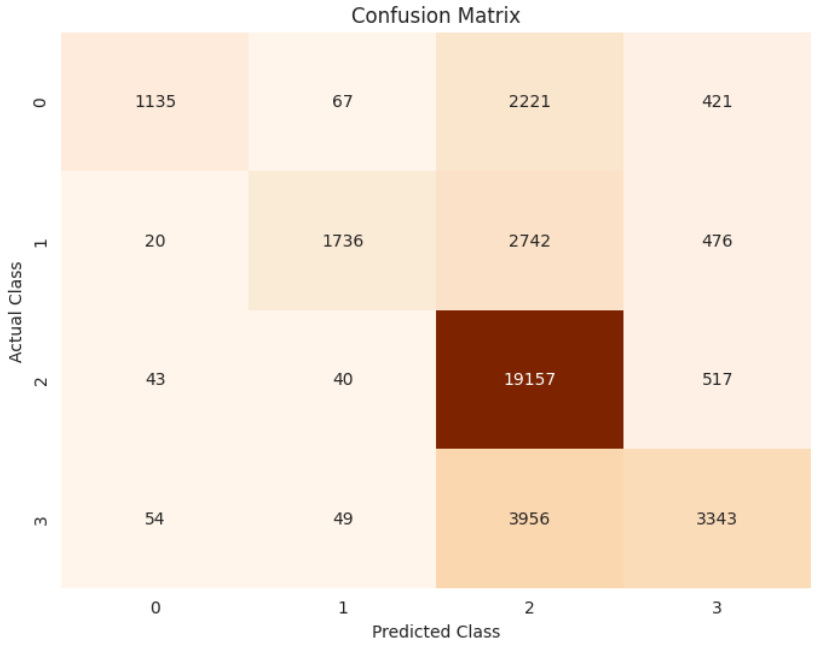
Decision Tree Confusion Matrix:



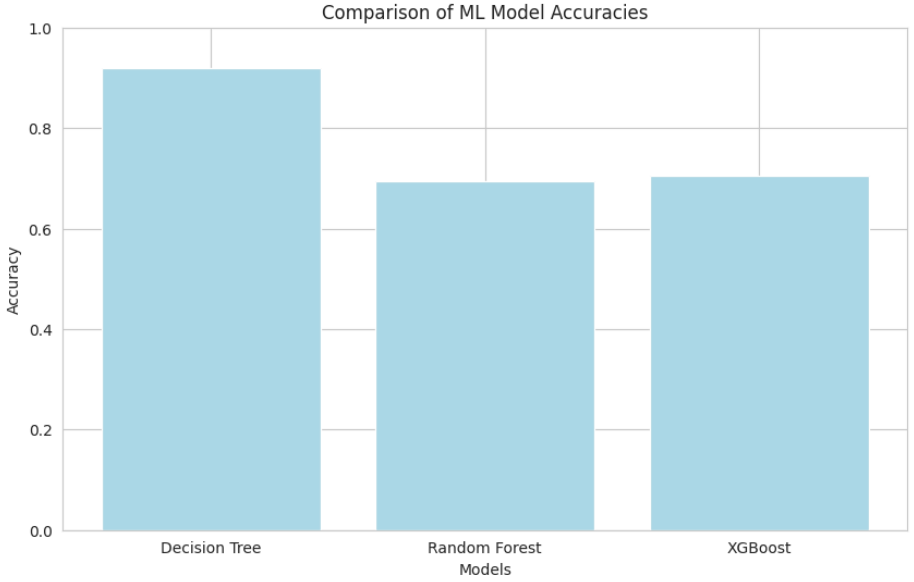
Random Forest Confusion Matrix:



XGBoost Confusion Matrix:



Comparison of accuracies of the models:



Shallow Neural Network:

Summary of the Neural Network:

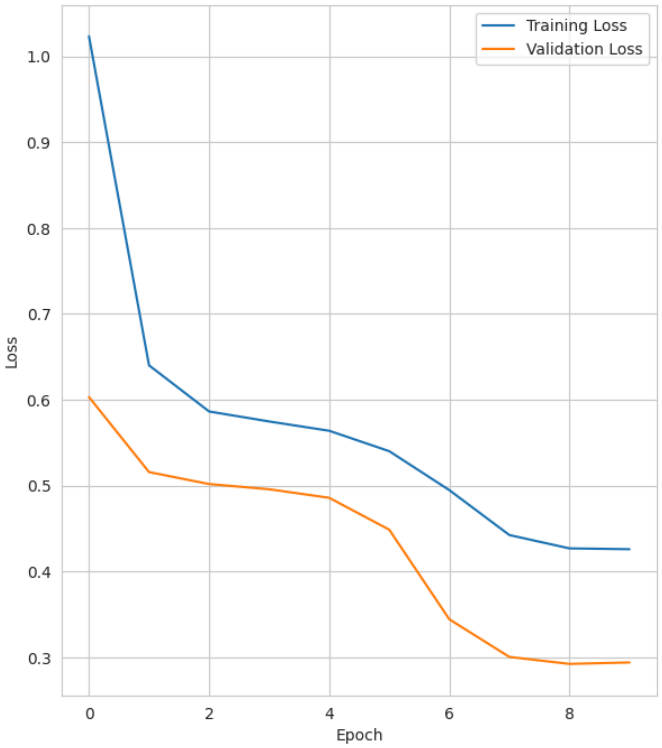
Layer (type)	Output Shape	Param #
Linear-1	[-1, 128]	40,064
Dropout-2	[-1, 128]	0
Linear-3	[-1, 64]	8,256
Dropout-4	[-1, 64]	0

```
            Linear-5                        [-1, 4]               260
================================================================
Total params: 48,580
Trainable params: 48,580
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.19
Estimated Total Size (MB): 0.19
----------------------------------------------------------------
None
```

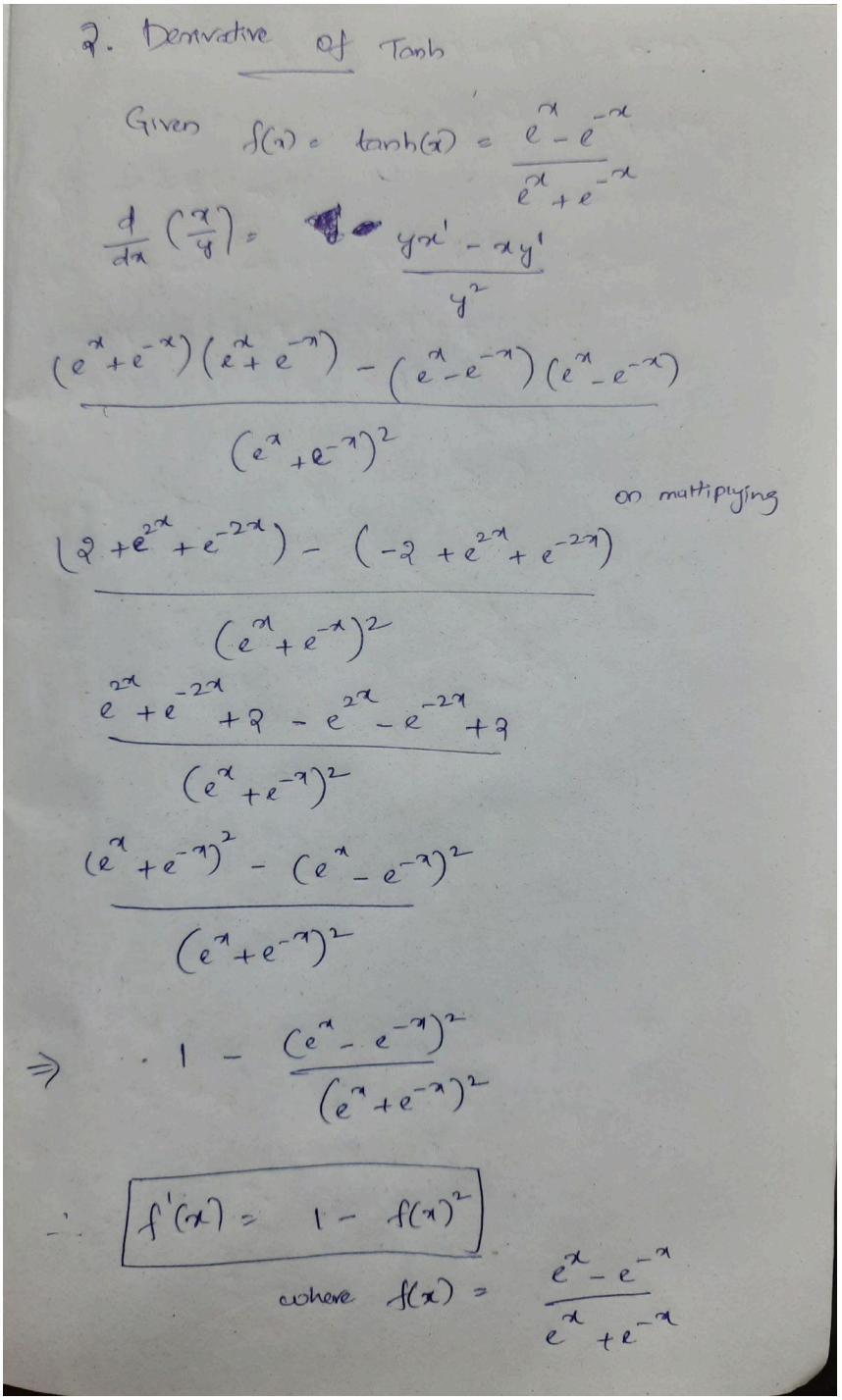
The training loss validation loss for running 10 Epochs are

Train Loss: 0.42
Validation Loss: 0.294
Accuracy: 86 %

And the below graph represents a comparison between Training loss and Validation loss. As we increase the number of Epochs the training and validation losses go on decreasing and achieved a accuracy of 86%.



2. Derivative of Tanh [5 points]



Part II: Deep Learning Theoretical Part [20 points] 1. Forward-backward Pass [15 points]

Look into the attached pdf

Forward Pass

1] $h_1 = 0.7 \times (-0.3) + 0.5 \times 0.15 + 0$?

$= 0.765$

$h_2 = 0.7 \times 0.8 + 0.5 \times 0.2 - 0.14$

$= 0.52$

$\hat{y} = 0.765 \times 0.7 + 0.52 \times 0.25 - 0.1$

$= 0.5655$

2] $MSE = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(0.5 - 0.5655)^2$

$= 0.002145$

3] Calculating the gradient using back-propagation,

$$w_1 = w_1 - \alpha \left[\frac{\partial \text{Error}}{\partial w_1} \right]$$

$$\frac{\partial \text{Error}}{\partial w_1} = \frac{\partial \text{Error}}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_1} \cdot \frac{\partial h_1}{\partial w_1}$$

$$\Rightarrow \frac{\frac{1}{2}(y - \hat{y})^2}{\partial \hat{y}} \cdot \frac{\partial (h_1 w_7 + h_2 w_8 + h_3 w_9)}{\partial h_1} \cdot \frac{\partial (i_1 w_1 + i_2 w_3 + i_3 w_5)}{\partial w_1}$$

$$\Rightarrow 2 \cdot \frac{1}{2}(y - \hat{y}) \cdot \frac{\partial (y - \hat{y})}{\partial y} \quad (w_7 \cdot i_1)$$

$$\frac{d\text{Error}}{dw_1} = (y-\hat{y})(-1)(w_7 \cdot i_1)$$

$$\Rightarrow w_2 = w_2 - \alpha\left(\frac{d\text{Error}}{dw_2}\right)$$

$$\frac{d\text{Error}}{dw_2} = \frac{d\text{Error}}{d\hat{y}} \cdot \frac{d\hat{y}}{dh_2} \cdot \frac{dh_2}{dw_2}$$

$$\frac{1}{2}(y-\hat{y})^2 \cdot \frac{d(h_1w_7 + h_2w_8 + h_3w_9)}{dh} \cdot \frac{i_1w_2 + i_2w_4 + i_3w_6}{dw_2}$$

$$2 \cdot \frac{1}{2}(y-\hat{y}) \cdot \frac{d(y-\hat{y})}{d\hat{y}} \cdot w_8 i_1$$

$$\Rightarrow -(y-\hat{y})\, w_8 i_1$$

$$w_3 = w_3 - \alpha\left(\frac{d\text{Error}}{dw_3}\right)$$

$$\frac{d\text{Error}}{dw_3} = \frac{d\text{Error}}{d\hat{y}} \cdot \frac{d\hat{y}}{dh_1} \cdot \frac{dh_1}{dw_3}$$

$$= \frac{1}{2}(y-\hat{y})^2 \cdot \frac{d(h_1w_7 + h_2w_8 + h_3w_9)}{dh_1} \cdot \frac{d(i_1w_1 + i_2w_3 + i_3w_5)}{dw_3}$$

$$2 \cdot \frac{1}{2}(y-\hat{y}) \cdot \frac{d(y-\hat{y})}{d\hat{y}} \cdot w_7 \cdot i_2$$

$$= -(y-\hat{y}) \cdot (w_7 i_2)$$

$$w_4 = w_4 - \alpha \left(\frac{d\text{Error}}{dw_4}\right)$$

$$\frac{d\text{Error}}{dw_4} = \frac{d\,\text{Error}}{d\hat{y}} \cdot \frac{d\hat{y}}{dh_2} \cdot \frac{dh_2}{dw_4}$$

$$\Rightarrow \frac{1}{2}(y-\hat{y})^2 \cdot \frac{d(h_1 w_7 + h_2 w_8 + h_3 w_9)}{dh_2} \cdot \frac{d(i_1 w_2 + i_2 w_4 + i_3 w_6)}{dw_4}$$

$$\Rightarrow 2\,\frac{1}{2}(y-\hat{y}) \cdot \frac{d(y-\hat{y})}{d\hat{y}} \cdot w_8 i_2$$

$$\Rightarrow -(y-\hat{y})\, w_8 i_2$$

$$w_5 = w_5 - \alpha \left(\frac{d\text{Error}}{dw_5}\right)$$

$$\frac{d\text{Error}}{dw_5} = \frac{d\text{Error}}{d\hat{y}} \cdot \frac{d\hat{y}}{dh_1} \cdot \frac{dh_1}{dw_5}$$

$$= \frac{1}{2}(y-\hat{y})^2 \cdot \frac{d(h_1 w_7 + h_2 w_8 + h_3 w_9)}{dh_1} \cdot \frac{d(i_1 w_1 + i_2 w_3 + i_3 w_5)}{dw_5}$$

$$-(y-\hat{y})\, w_7 i_3$$

$$w_6 = w_6 - \alpha \left(\frac{d\text{Error}}{dw_6}\right)$$

$$\frac{d\text{Error}}{dw_6} = \frac{d\text{Error}}{d\hat{y}} \cdot \frac{d\hat{y}}{dh_2} \cdot \frac{dh_2}{dw_6}$$

$$= \frac{1}{2}(y-\hat{y})^2 \cdot \frac{d(h_1 w_7 + h_2 w_8 + h_3 w_9)}{dh_2} \cdot \frac{d(i_1 w_2 + i_2 w_4 + i_3 w_6)}{dw_6}$$

$= -(y-\hat{y}) w_8 i_3$

$w_7 = w_7 - \alpha \left(\frac{\partial Error}{\partial w_7}\right)$

$\frac{\partial Error}{\partial w_7} = \frac{\partial Error}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_7}$

$= \frac{1}{2}(y-\hat{y})^2 \cdot \frac{\partial (h_1 w_7 + h_2 w_8 + h_3 w_9)}{\partial w_7}$

$= -(y-\hat{y}) \cdot h_1$

$w_8 = w_8 - \alpha \left(\frac{\partial Error}{\partial w_8}\right)$

$\frac{\partial Error}{\partial w_8} = \frac{\partial Error}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_8}$

$\frac{1}{2}(y-\hat{y})^2 \cdot \frac{\partial (h_1 w_7 + h_2 w_8 + h_3 w_9)}{\partial w_8}$

$-(y-\hat{y}) h_2$

$w_9 = w_9 - \alpha \left(\frac{\partial Error}{\partial w_9}\right)$

$\frac{\partial Error}{\partial w_9} = \frac{\partial Error}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_9}$

$\frac{1}{2}(y-\hat{y})^2 \cdot \frac{\partial (h_1 w_7 + h_2 w_8 + h_3 w_9)}{\partial w_9}$

$-(y-\hat{y}) \cdot h_3$

after Calculating weights we obtained as.

$$w_9 = w_9 - \alpha(-h_3(y-\hat{y}))$$

$$w_8 = w_8 - \alpha(-h_2(y-\hat{y}))$$

$$w_7 = w_7 - \alpha(-h_1(y-\hat{y}))$$

$$w_6 = w_6 - \alpha(-i_3 w_8 (y-\hat{y}))$$

$$w_5 = w_5 - \alpha(-i_3 w_7 (y-\hat{y}))$$

$$w_4 = w_4 - \alpha(-i_2 w_8 (y-\hat{y}))$$

$$w_3 = w_3 - \alpha(-i_2 (y-\hat{y}) w_7)$$

$$w_2 = w_2 - \alpha(-i_1 (y-\hat{y}) w_8)$$

$$w_1 = w_1 - \alpha(-i_1 (y-\hat{y}) w_7))$$

Calculating the updated weights we obtain

$$w_1 = -0.3 - 0.03(0.7 \times 0.7 \times 0.0655)$$

$$= -0.3009$$

$$w_2 = 0.8 - 0.03(0.7 \times 0.25 \times 0.0655)$$

$$= 0.7996$$

$$w_3 = 0.15 - 0.03(0.5 \times 0.7 \times 0.0655)$$

$$= 0.1493$$

$$w_4 = 0.2 - 0.03(0.5 \times 0.25 \times 0.0655)$$

$$= 0.1997$$

$w_5 = 0.9 - 0.03\ (1 \times 0.7 \times 0.0655)$

$= 0.8986$

$w_6 = -0.14 - 0.03\ (1 \times 0.25 \times 0.0655)$

$= -0.1404$

$w_7 = 0.7 - 0.03\ (0.765 \times 0.0655)$

$= 0.6984$

$w_8 = 0.25 - 0.03\ (0.52 \times 0.0655) = 0.2489$

$w_9 = -0.1 - 0.03\ (1 \times 0.0655) = -0.1019$

$h_1 = 0.7 \times -0.3009$

$+ 0.5 \times 0.1493$

$+ 1 \times 0.8586 \Rightarrow 0.7626$

$h_2 = 0.7 \times 0.7996$

$+ 0.5 \times 0.1997$

$+ 1 \times (-0.1404) \Rightarrow 0.51917$

$\hat{y} = 0.7626 \times 0.6984 + 0.519 \times 0.24 - 0.1019$

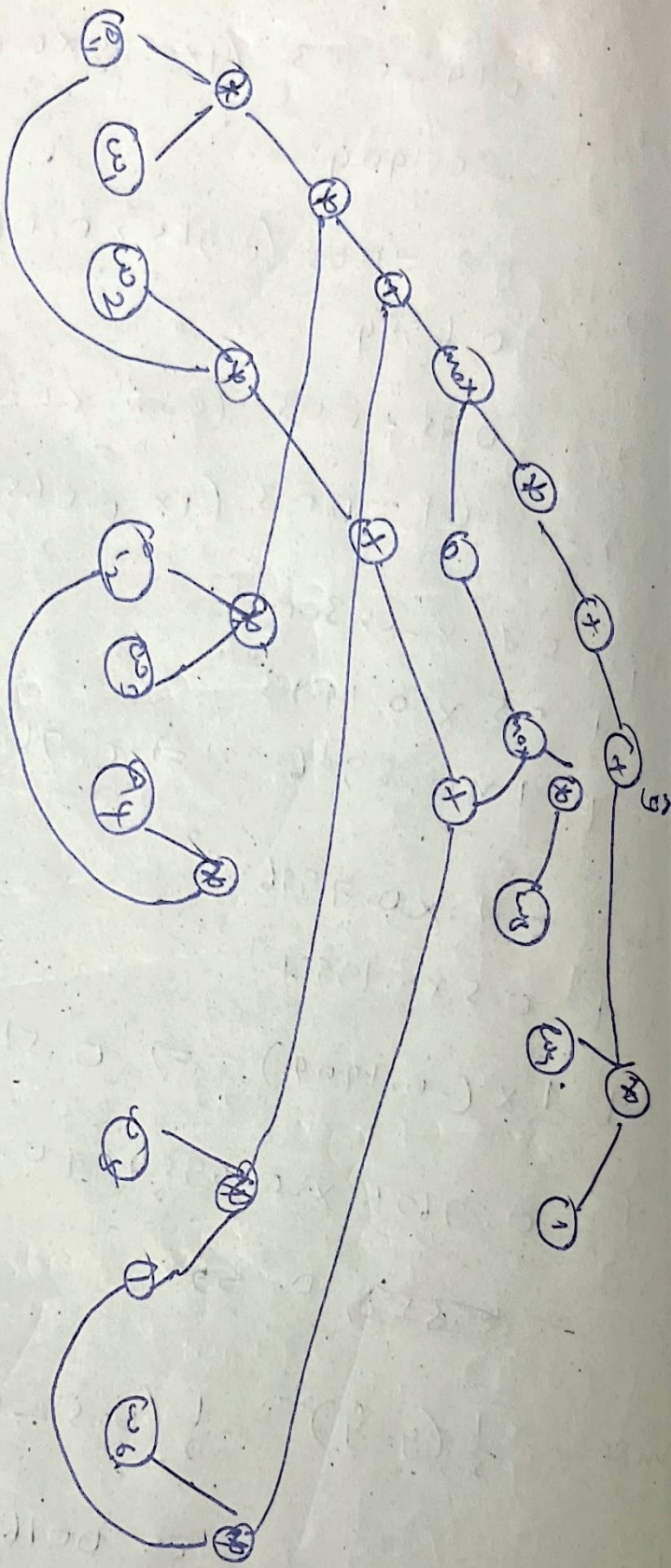
$= 0.558$

$MSE = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(0.5 - 0.558)^2$

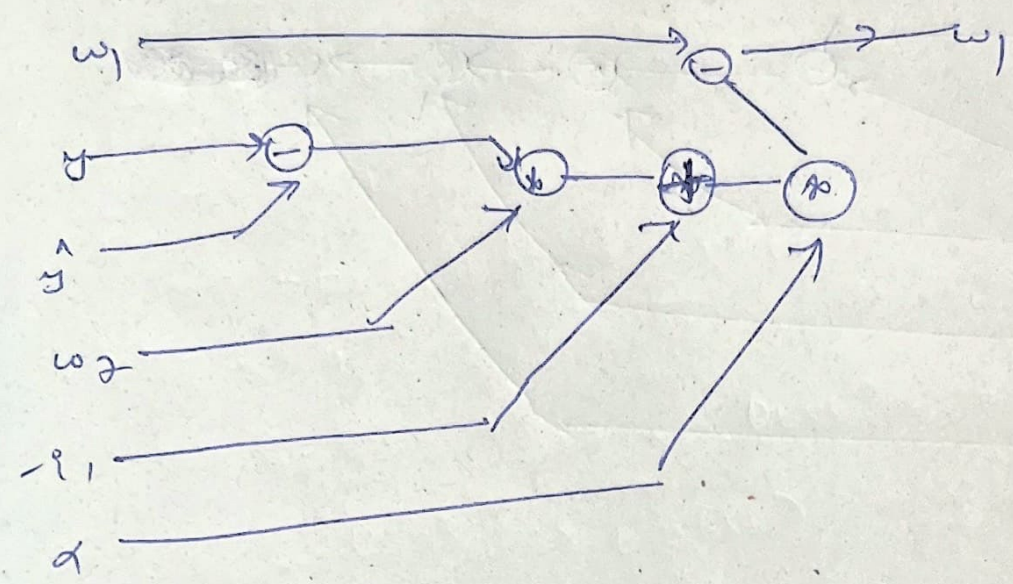
$= 0.001682$

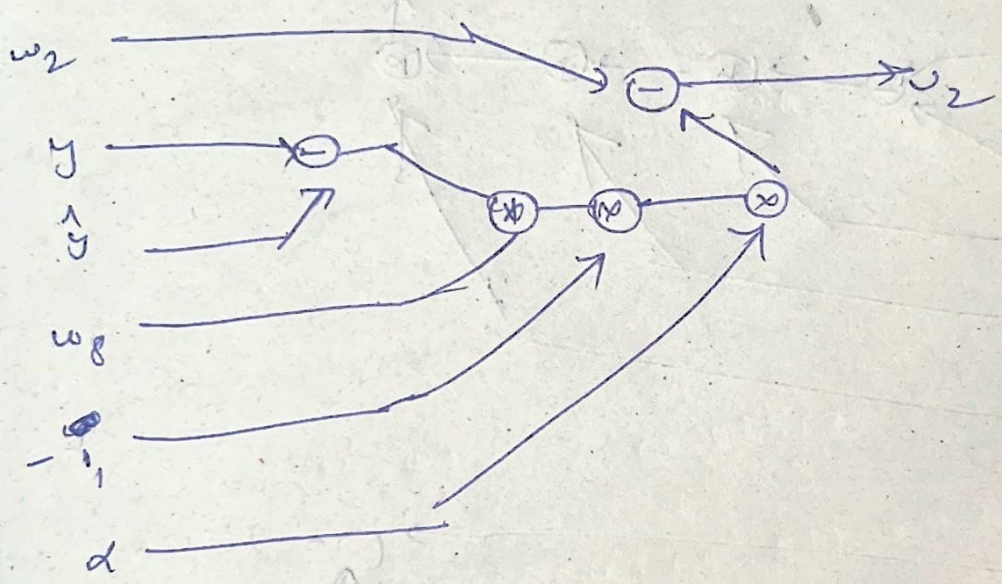
So mse from 1st forward pass is 0.0022 and 2nd forward pass is 0.001684. That means the error is reduced

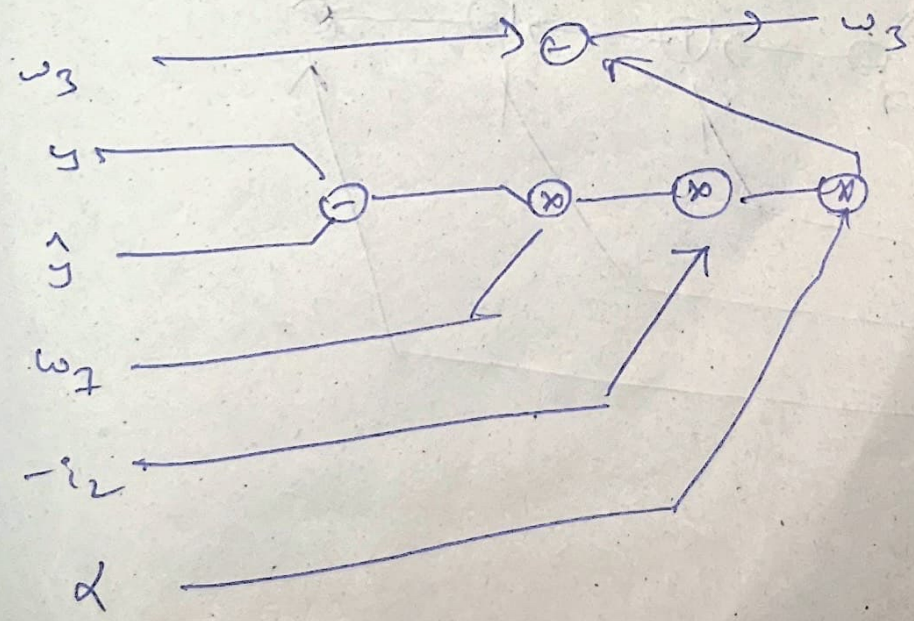
Forward Pass

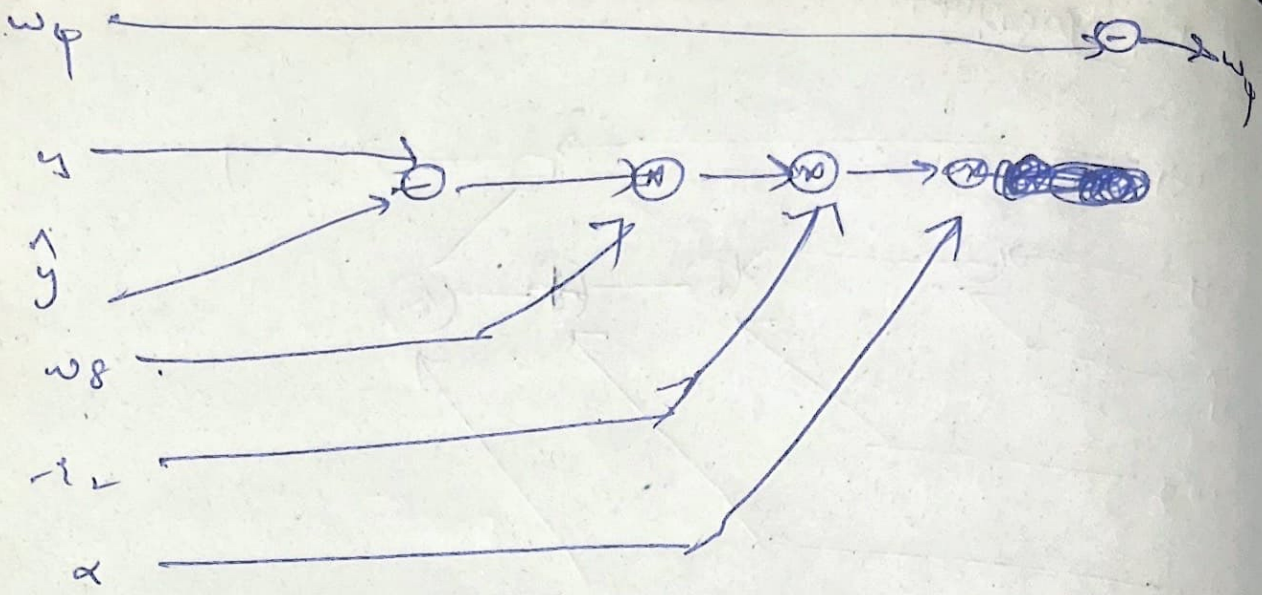


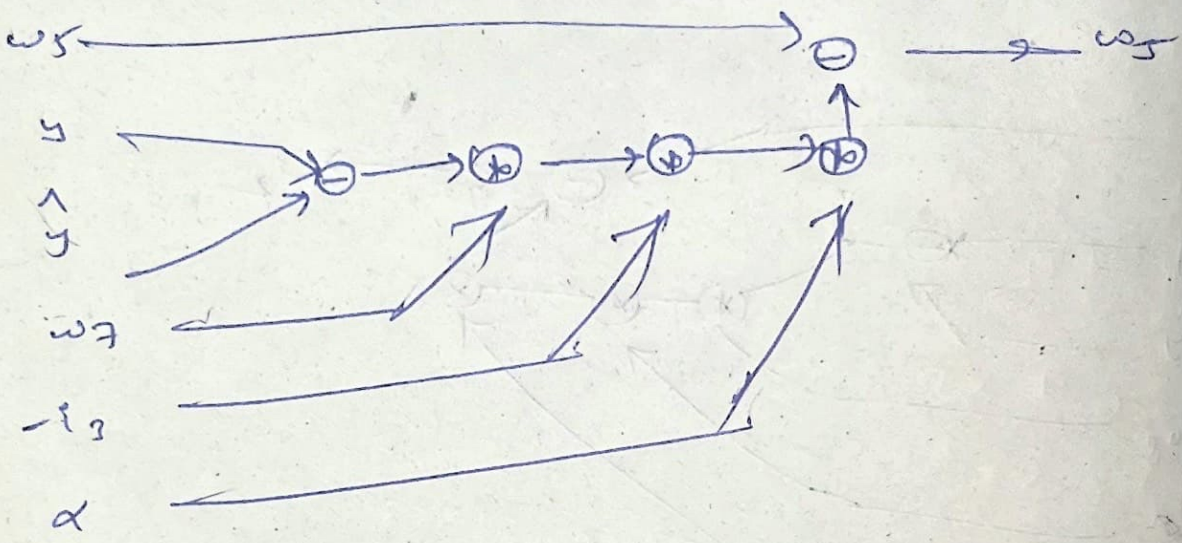
Back Propagation

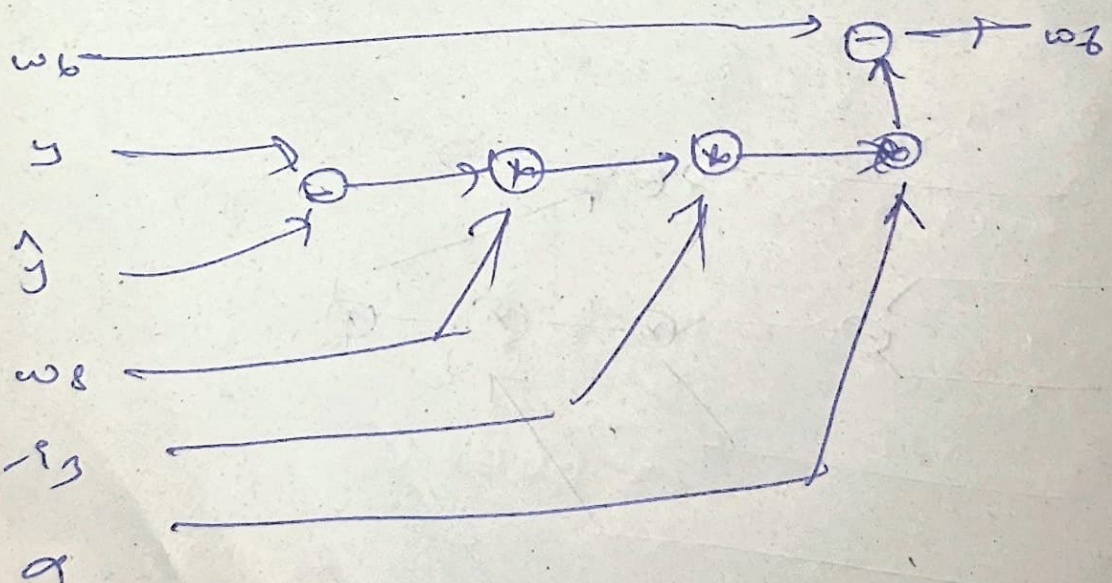


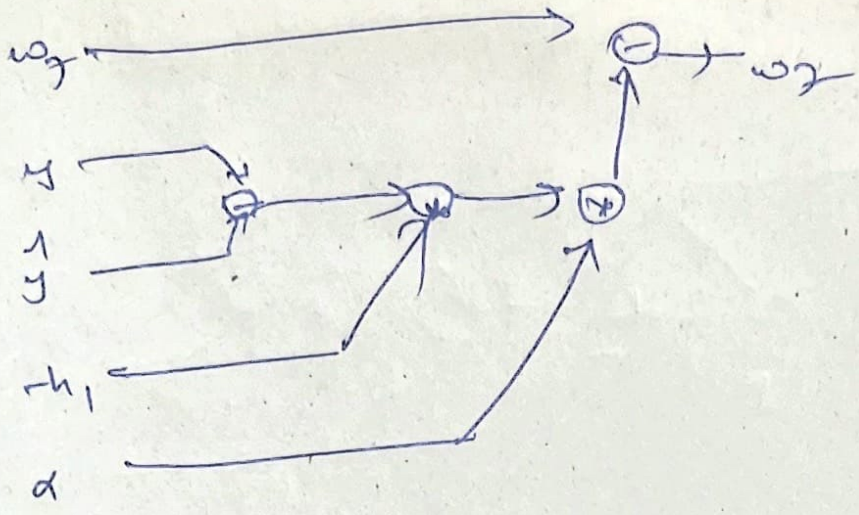


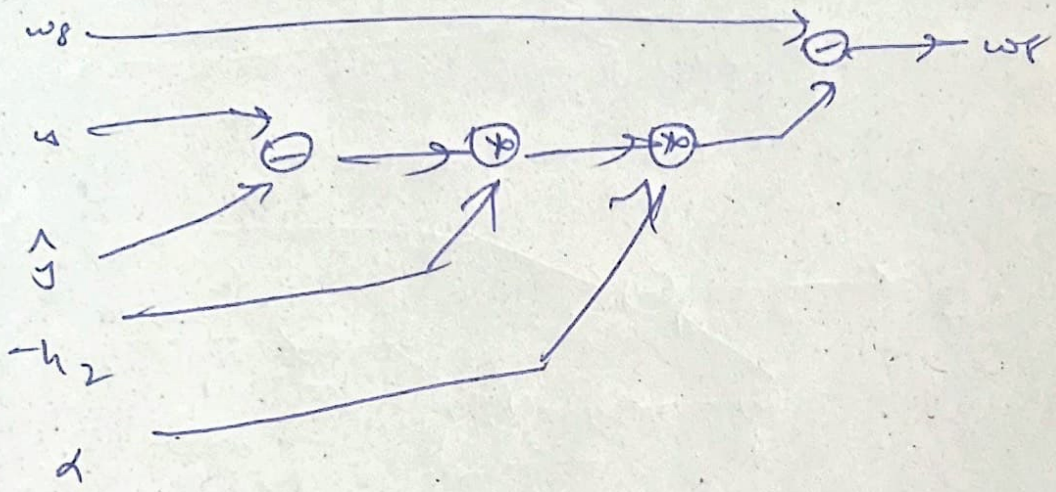


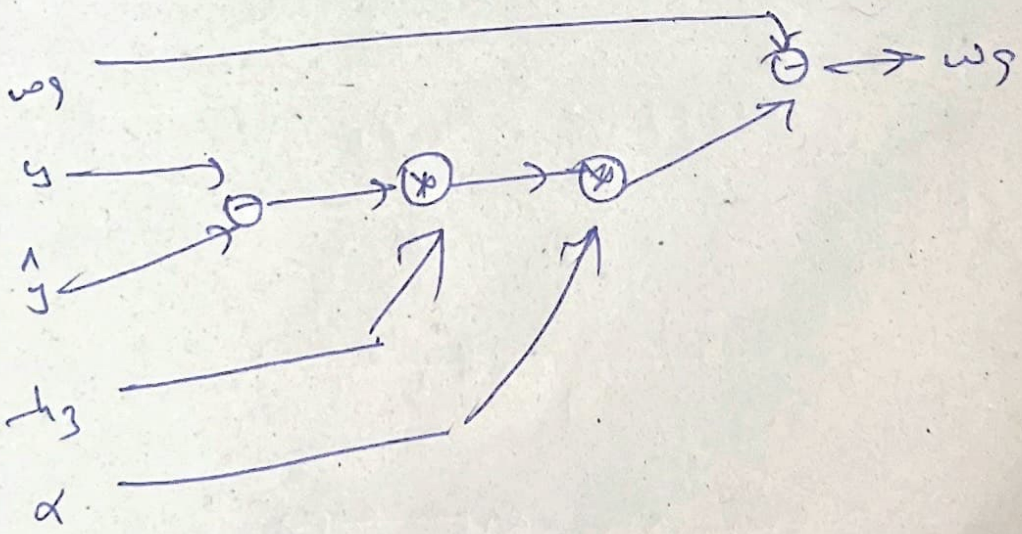

w_9
w_9
y
$\hat{y}$
w_8
$-i_L$
α


w_5
w_5
y
$\hat{y}$
w_7
$-i_3$
α


w_6
w_6
y
$\hat{y}$
w_8
$-i_3$
α


w_7
w_2
y
$\hat{y}$
$-h_1$
α


w_8
w_4
y
$\hat{y}$
$-h_2$
α


w_9
w_5
y
$\hat{y}$
h_3
α

Part III: OCTMNIST Classification:

OCTMNIST Dataset:

Number of classes in the dataset: 4
Length of combined dataset 109309
Length of train_size: 76516
Length of val_size: 16396
Length of test_size: 16397

Base Model:

Summary of the defined Neural Network:

```
modelCNN(
  (convlayers): Sequential(
    (0): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fclayers): Sequential(
    (0): Linear(in_features=1568, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=36, bias=True)
  )
)
```

```
----------------------------------------------------------------
      Layer (type)               Output Shape         Param #
================================================================
          Conv2d-1         [-1, 16, 28, 28]             160
            ReLU-2         [-1, 16, 28, 28]               0
       MaxPool2d-3         [-1, 16, 14, 14]               0
          Conv2d-4         [-1, 32, 14, 14]           4,640
            ReLU-5         [-1, 32, 14, 14]               0
       MaxPool2d-6           [-1, 32, 7, 7]               0
          Linear-7                [-1, 128]         200,832
            ReLU-8                [-1, 128]               0
          Linear-9                 [-1, 36]           4,644
================================================================
Total params: 210,276
Trainable params: 210,276
Non-trainable params: 0
```

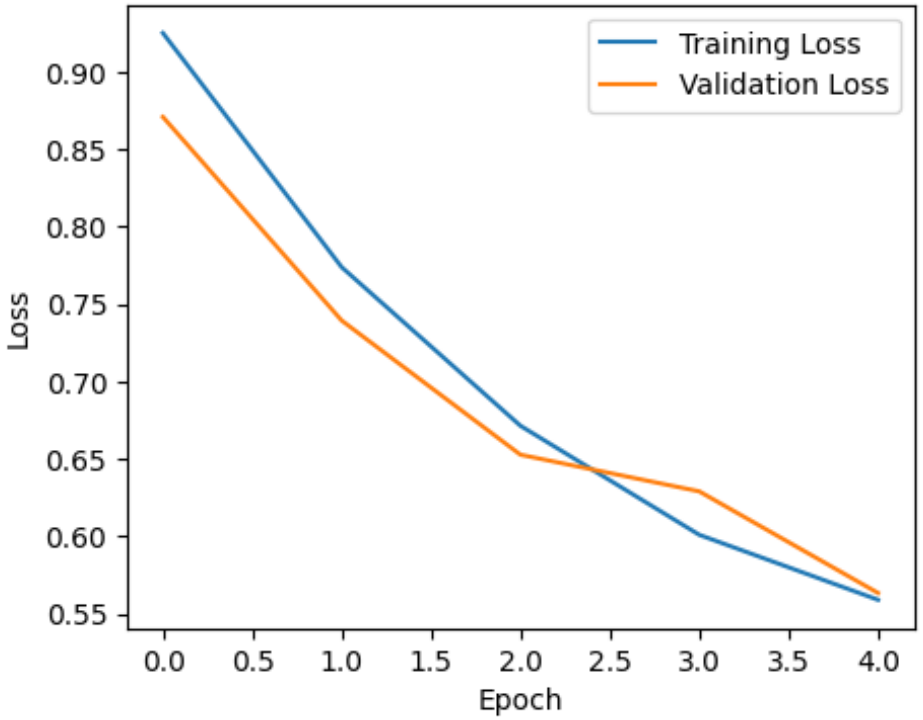
Input size (MB): 0.00
Forward/backward pass size (MB): 0.33
Params size (MB): 0.80
Estimated Total Size (MB): 1.13

Model Parameters:

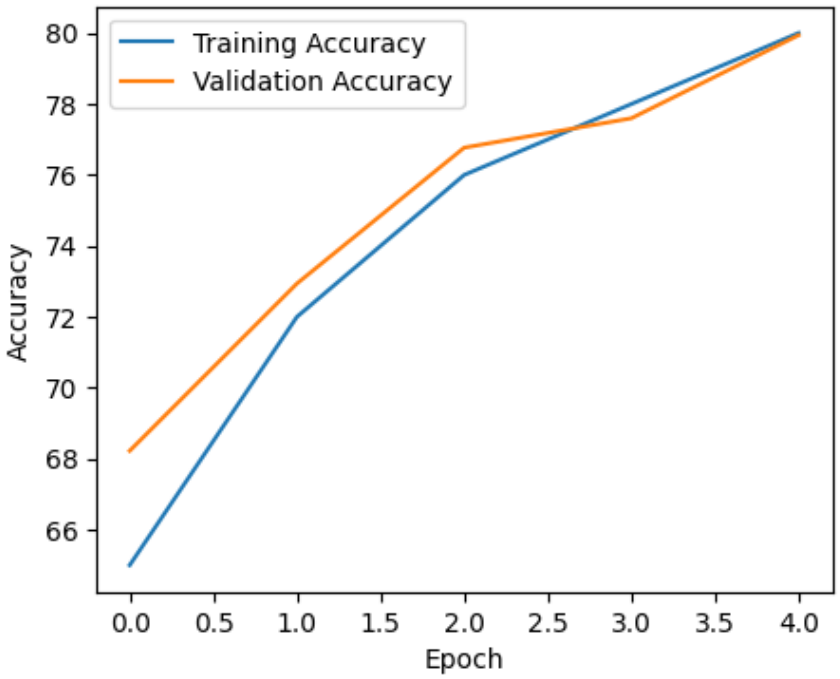
Optimizer: SGD
Learning rate: 0.004
Loss function: CrossEntropyLoss()

	Accuracy	Loss
Training	80.00%	0.55
Validation	79.94%	0.56
Testing	80%	0.55

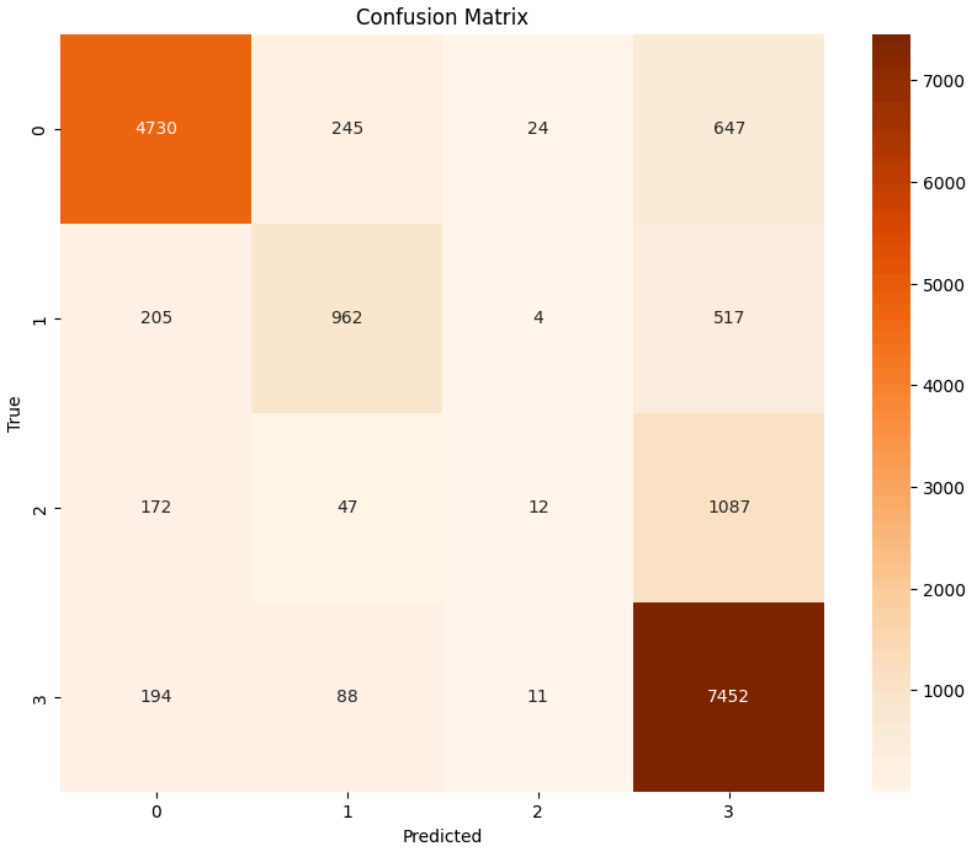
Training and Validation Loss Plot:



Training and Validation Loss:



Confusion Matrix:



Model Metrics:

Precision: 0.76
Recall: 0.80
F1 Score: 0.7

Optimising Methods:

Early Stopping:

Summary of the Model:

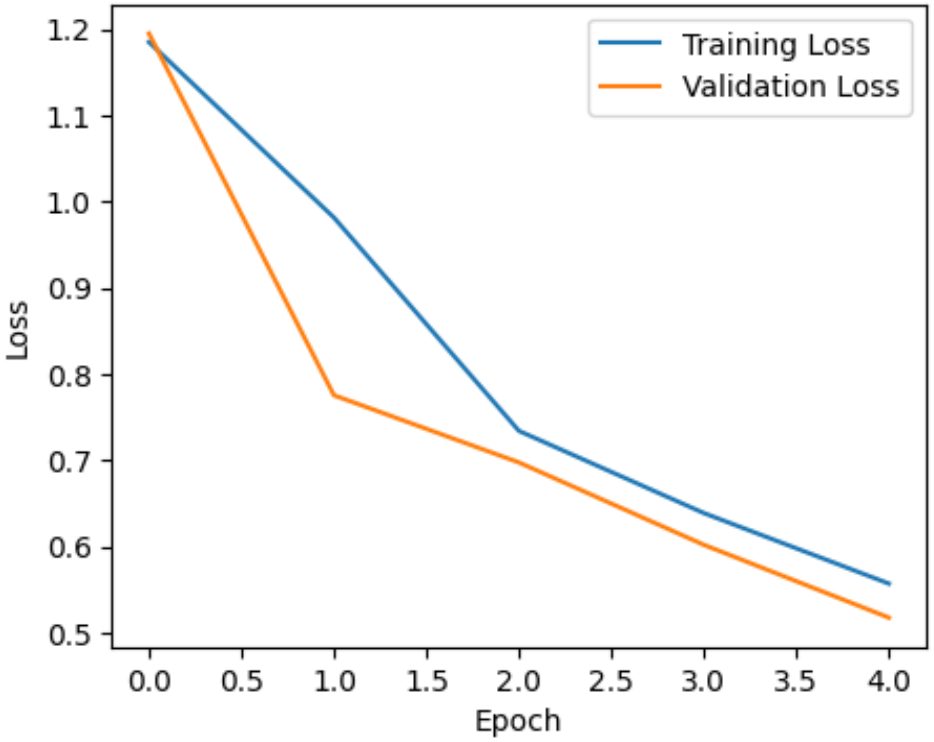
```
modelCNN(
  (convlayers): Sequential(
    (0): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fclayers): Sequential(
    (0): Linear(in_features=1568, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=36, bias=True)
  )
)
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 16, 28, 28]             160
              ReLU-2         [-1, 16, 28, 28]               0
         MaxPool2d-3         [-1, 16, 14, 14]               0
            Conv2d-4         [-1, 32, 14, 14]           4,640
              ReLU-5         [-1, 32, 14, 14]               0
         MaxPool2d-6           [-1, 32, 7, 7]               0
            Linear-7                [-1, 128]         200,832
              ReLU-8                [-1, 128]               0
            Linear-9                 [-1, 36]           4,644
================================================================
Total params: 210,276
Trainable params: 210,276
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.33
Params size (MB): 0.80
Estimated Total Size (MB): 1.13
```

Parameters:

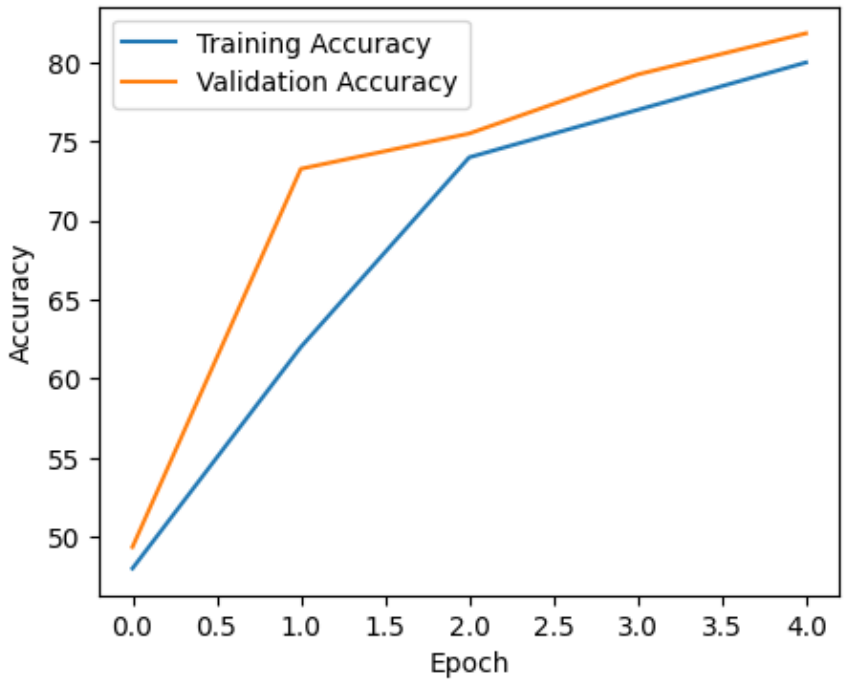
Learning Rate: 0.004
Optimizer: SGD
Loss Function: CrossEntropyLoss()

	Accuracy	Loss
Training	80.00%	0.55
Validation	81.84%	0.51
Testing	81%	0.52

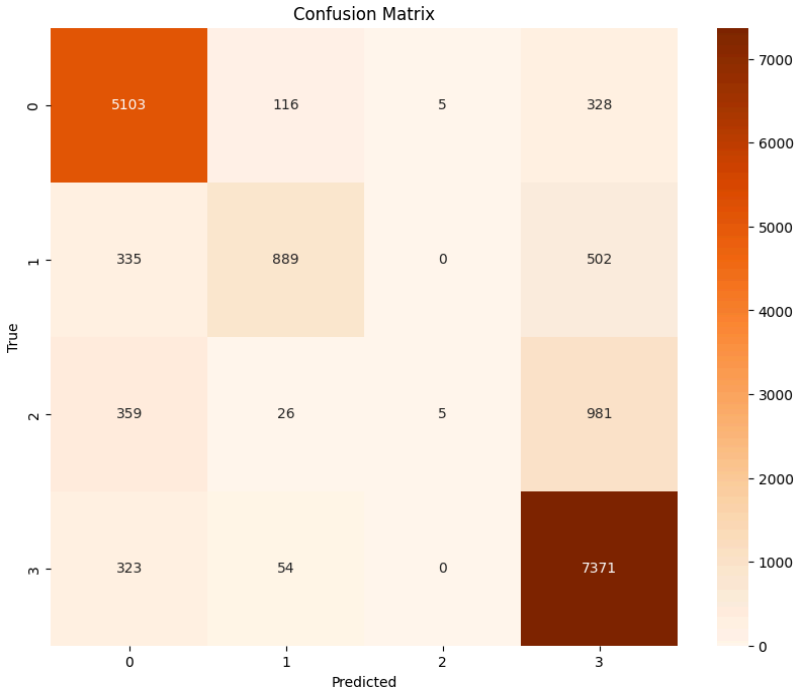
Plots for Training loss and validation loss:



Plot for Training and Validation Accuracy:



Confusion Matrix:



Model Metrics

Precision: 0.79

Recall: 0.82
F1 Score: 0.77

Dropout:

Summary of the model:

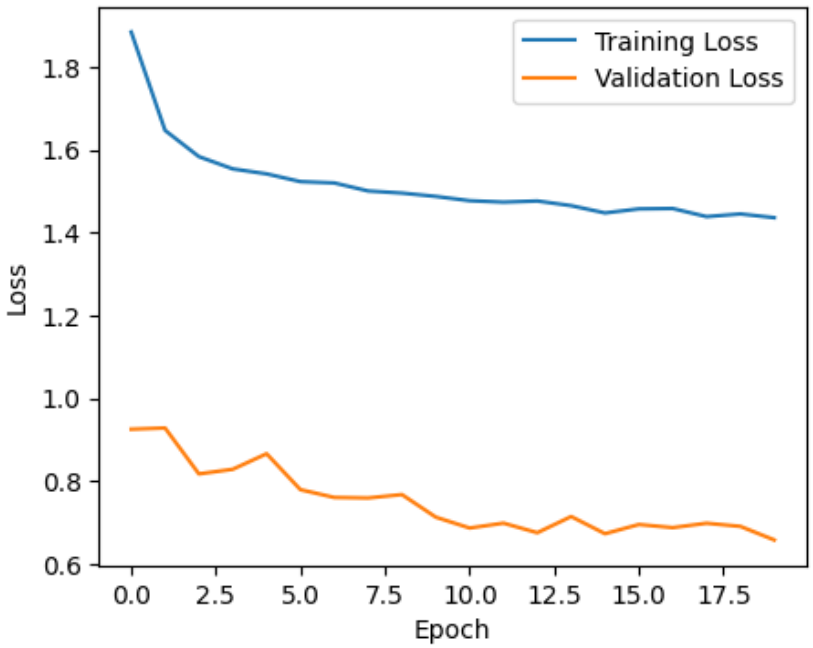
```
modelCNN(
  (convlayers): Sequential(
    (0): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fclayers): Sequential(
    (0): Linear(in_features=1568, out_features=128, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=128, out_features=36, bias=True)
    (4): ReLU()
    (5): Dropout(p=0.5, inplace=False)
  )
)
----------------------------------------------------------------
      Layer (type)               Output Shape         Param #
================================================================
          Conv2d-1           [-1, 16, 28, 28]             160
            ReLU-2           [-1, 16, 28, 28]               0
         MaxPool2d-3           [-1, 16, 14, 14]               0
          Conv2d-4           [-1, 32, 14, 14]           4,640
            ReLU-5           [-1, 32, 14, 14]               0
         MaxPool2d-6             [-1, 32, 7, 7]               0
          Linear-7                  [-1, 128]         200,832
            ReLU-8                  [-1, 128]               0
         Dropout-9                  [-1, 128]               0
         Linear-10                   [-1, 36]           4,644
           ReLU-11                   [-1, 36]               0
        Dropout-12                   [-1, 36]               0
================================================================
Total params: 210,276
Trainable params: 210,276
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.33
Params size (MB): 0.80
Estimated Total Size (MB): 1.13
```

Parameters:

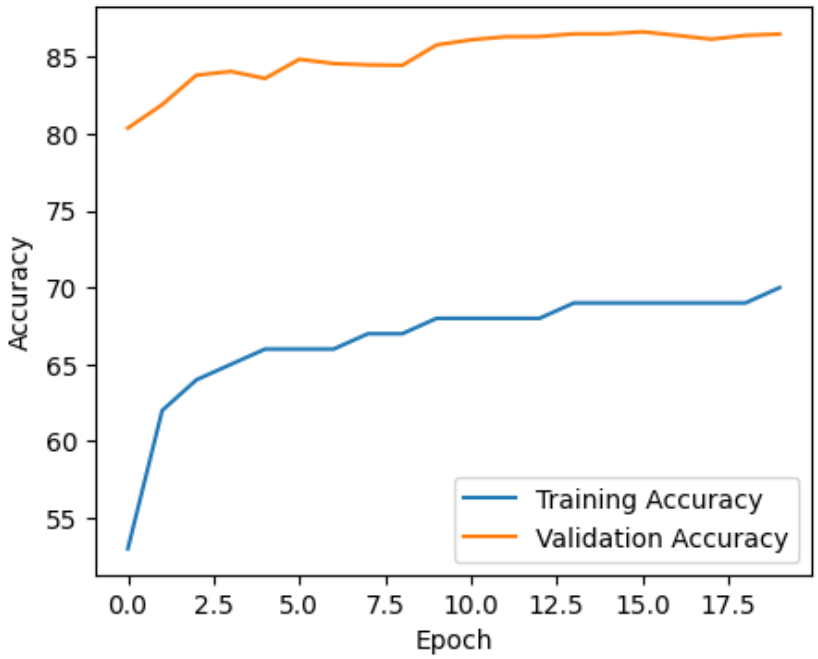
Learning Rate: 0.004
Optimizer: Adam
Loss Function: CrossEntropyLoss()
Dropout rate: 0.3

	Accuracy	Loss
Training	70.00%	1.43
Validation	86.51%	0.65
Testing	87%	0.64

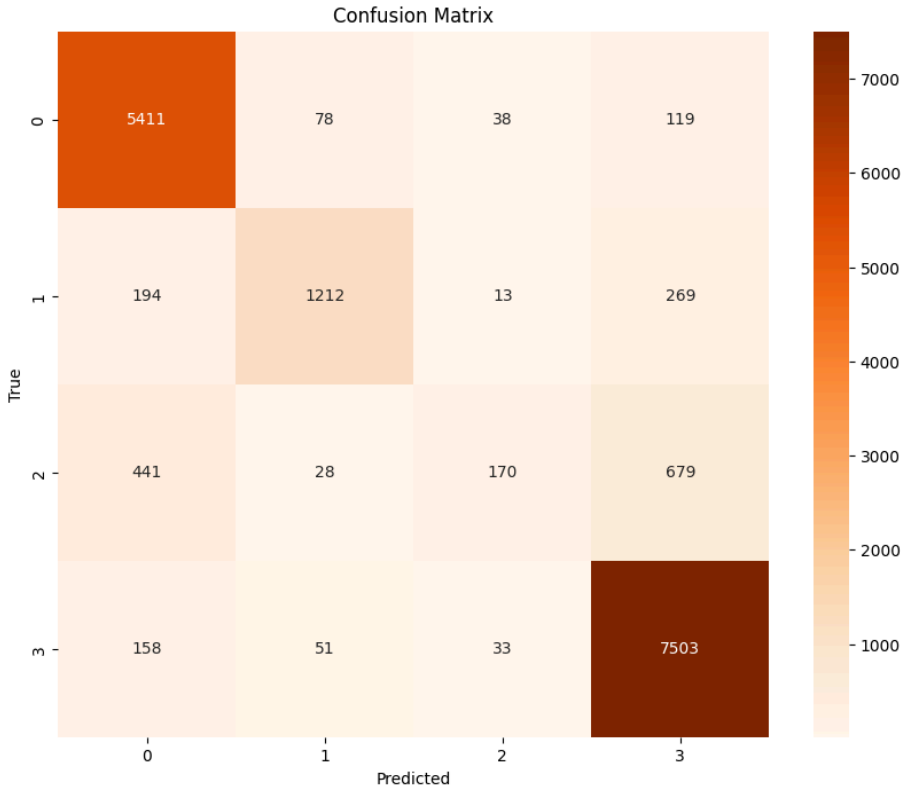
Plots for Training loss and validation loss:



Plot for Training and Validation Accuracy:



Confusion Matrix:



Model Metrics

Precision: 0.86
Recall: 0.87
F1 Score: 0.85

L2 regularization

Summary of the model:

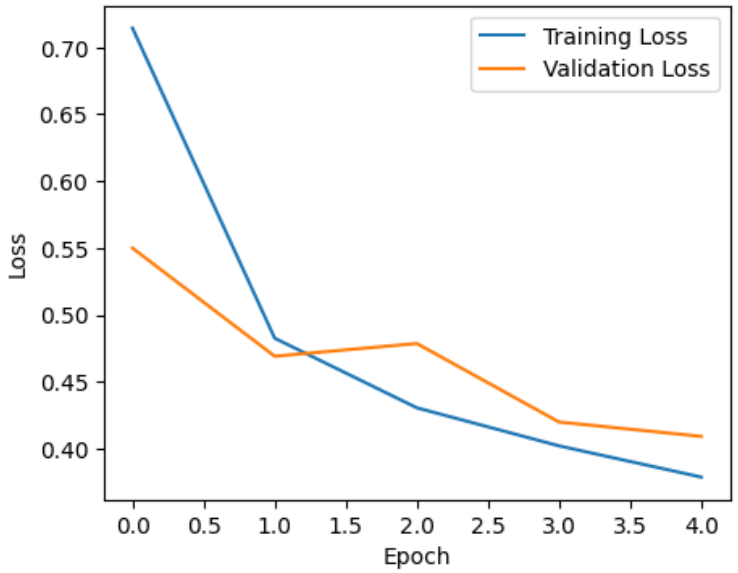
```
modelCNN(
  (convlayers): Sequential(
    (0): Conv2d(1, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (fclayers): Sequential(
    (0): Linear(in_features=1568, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=36, bias=True)
  )
)
----------------------------------------------------------------
      Layer (type)               Output Shape         Param #
================================================================
          Conv2d-1         [-1, 16, 28, 28]             160
            ReLU-2         [-1, 16, 28, 28]               0
       MaxPool2d-3         [-1, 16, 14, 14]               0
          Conv2d-4         [-1, 32, 14, 14]           4,640
            ReLU-5         [-1, 32, 14, 14]               0
       MaxPool2d-6           [-1, 32, 7, 7]               0
          Linear-7                [-1, 128]         200,832
            ReLU-8                [-1, 128]               0
          Linear-9                 [-1, 36]           4,644
================================================================
Total params: 210,276
Trainable params: 210,276
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.00
Forward/backward pass size (MB): 0.33
Params size (MB): 0.80
Estimated Total Size (MB): 1.13
----------------------------------------------------------------
```

Model Parameters:

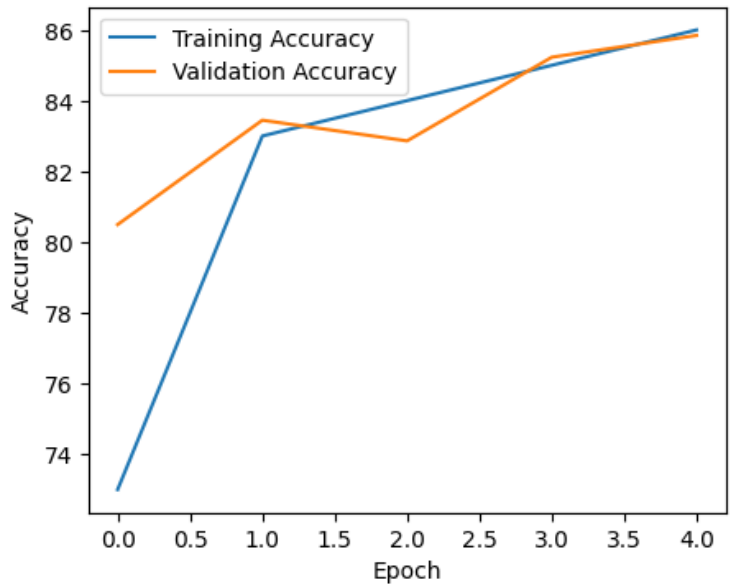
Learning Rate: 0.004
Optimizer: SGD
Loss Function: CrossEntropyLoss()
Weight_decay: 1e-5

	Accuracy	Loss
Training	86.00%	0.37
Validation	85.84%	0.40
Testing	85%	0.40

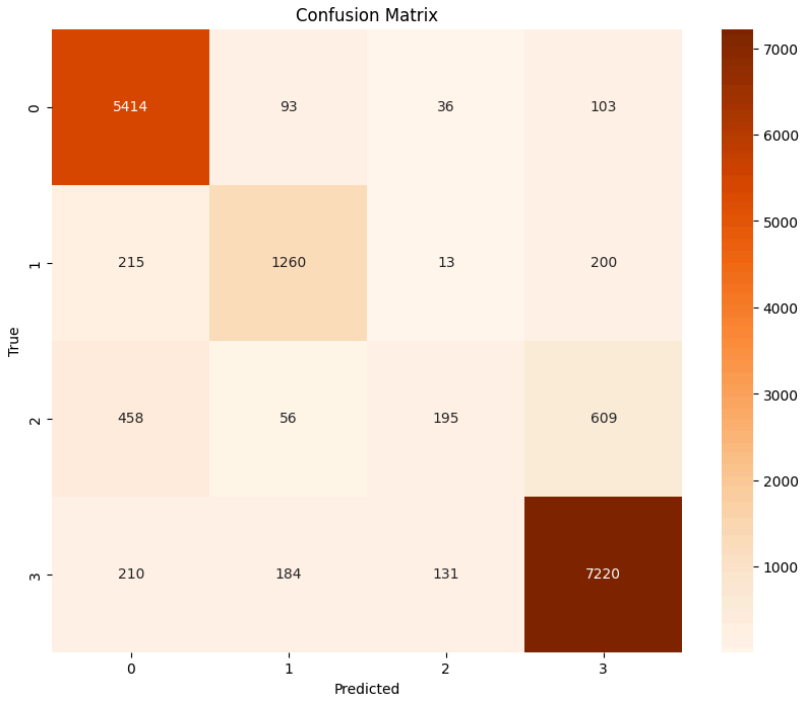
Plots for Training loss and validation loss:



Plot for Training and Validation Accuracy:



Confusion Matrix:



Evaluation Metrics of the model:

Precision: 0.84
Recall: 0.86
F1 Score: 0.84

Observations:

From the three optimization techniques: Early Stopping, adding Dropouts, and L2 Regularization. The three techniques work better than base model by increasing overall accuracy on other hand decreasing loss. They are preventing model from getting overfitted. There is an increase in approximately 7% accuracy from the base model. So when we observe, adding dropouts increases the accuracy of the model and prevents the model overfitting. Hence this technique best suits the given data.

References:

CSE 574 Machine Learning Assignment 1 submission by Dharma. Acha
CSE 574 Machine Learning Assignment 2 submission by Dharma. Acha
https://pandas.pydata.org/
https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html
https://en.wikipedia.org/wiki/Early_stopping
https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
https://scikit-learn.org/stable/
https://health.data.ny.gov/Health/New-York-State-Statewide-COVID-19-Hospitalizations/jw46-jpb7/about_data