CSE 676 : Deep Learning Assignment 0

Name : Priya Ramesh Patil
UB Pearson Number : 50544948
UBIT Name : ppatil22

PART I : Data Analysis, ML models & PyTorch

Step 1 : Data Analysis and Pre-processing :

Problem Statement : Predicting the fare amount for the trips made by Green Taxis in New York

About the Dataset :
The dataset *'2013_Green_Taxi_Trip_Data.csv'* sourced from https://www.data.gov/ this website contains information on the green taxi trips in New York City for the year 2013. This dataset contains all the details for every trip such as pickup and drop-off times, pickup and drop locations, trip distances, vendors running the taxis, no of passengers and trip fares. The rows in this dataset describe about the individual taxi trips for every customer where as the columns describe the specific trip details.
Statistics observed from the data are as follows :

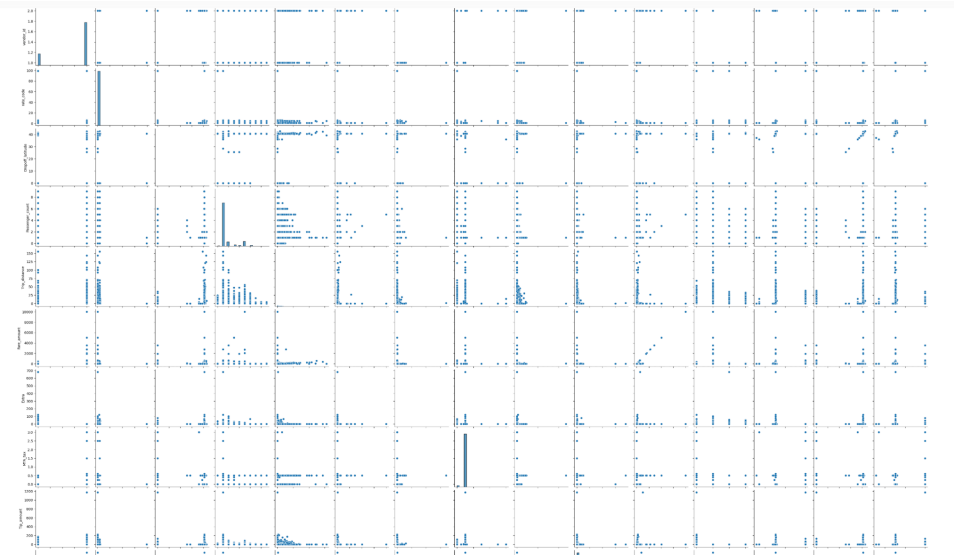
- Average trip distance id —miles and has a standard deviation of —
- Average fare amount for trip is — miles and has a standard deviation of —
- Maximum number of trips are made by vendor having id = 2

Pre-processing Techniques :

1. Check for missing values : After checking for any missing values present in the dataset , it was observed that a column only had NaN values. This column was then dropped from the data.
2. Check for Outliers : In order to see if there are any outliers box plot for every numerical feature was observed. On the observation there were no significant outliers present in any feature of the dataset
3. Handling Categorical Data : Transformation of Categorical data into numerical data using Label encoding for making further computations smooth.
4. Scaling Data : Scaled the data using StandardScaler() to make the data uniform within the range of (1,-1). The scaling technique helps to get more efficient results and improve the performance of the models.

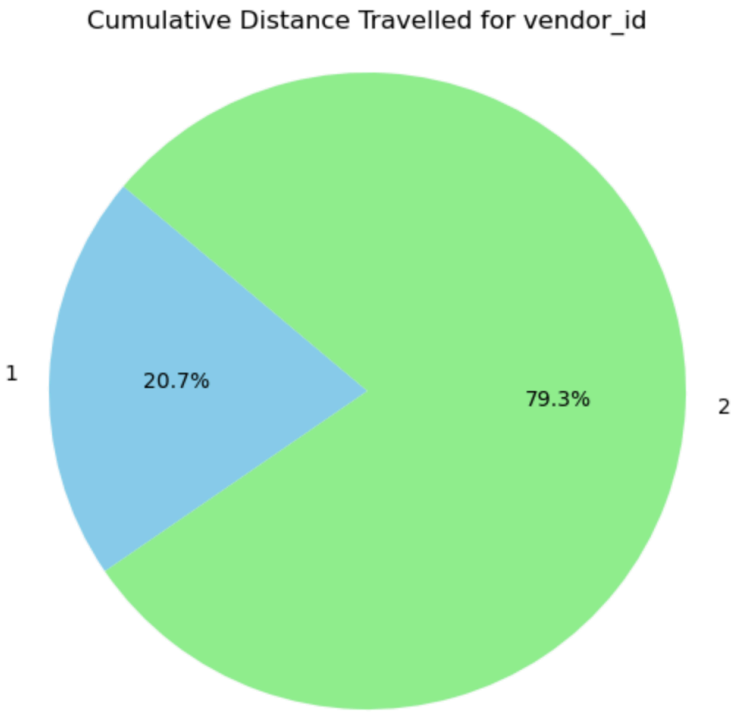
Exploratory Data Analysis :

1. **Pairwise Scatter Plot :**



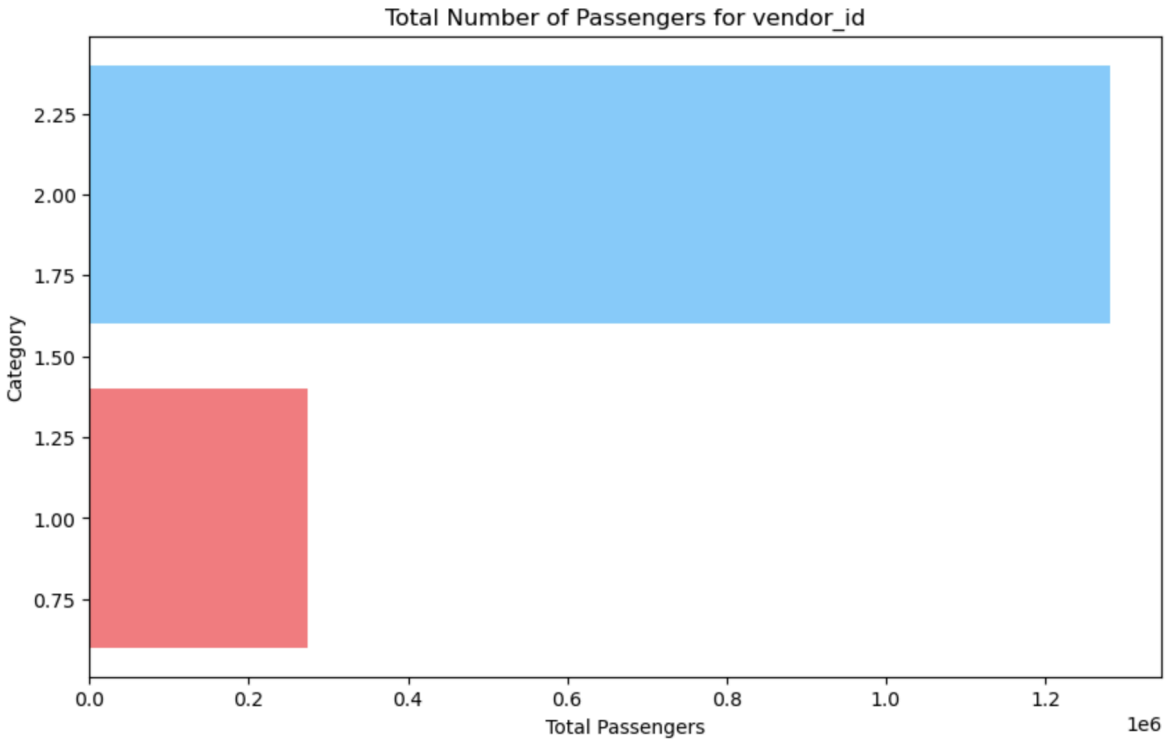
This plot describes the pairwise relationship between various features in the dataset. On observation of the plot there is no specific type of trend or relationship between the features. Although some of the features display some relationships among themselves but they don't posses strong co linearity .

2. Cumulative Distance travelled of by every vendor



This plot is a pie chart that describes the cumulative distance travelled by passengers through the vendor. The plot states that 79.3% of distance is travelled through the vendor 2 and 20.7% of distance is travelled through vendor 1. Clearly the most frequently operating vendor is vendor 2

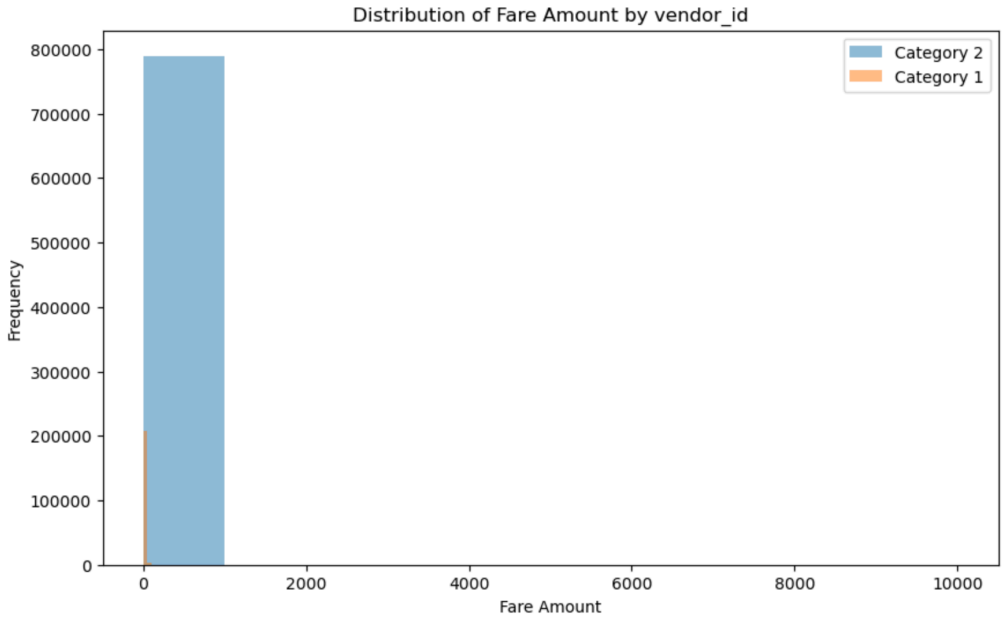
3. Total Number of passengers served by every vendor



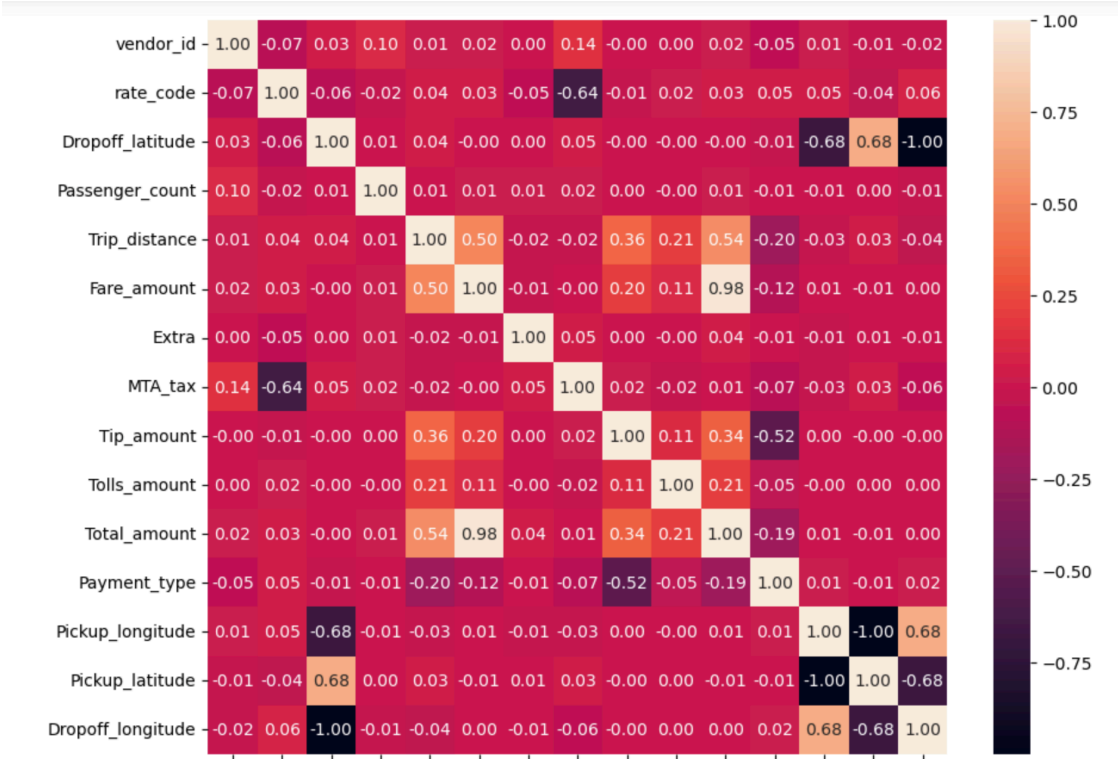
This plot is a bar chart that display the number of passengers served by every vendor. The bar for vendor 2 is pretty longer that the bar for vendor 1. this indicates that there are significantly more number of passengers preferring vendor 2.

4. Distribution of Fare amount

This plot describes the distribution of the target variable being predicted in this problem statement that is the fare amount for the trip. For every vendor category this is the distribution of the fare_amount.



5. Heatmap



This plot describes how strong a feature is correlated with other. The boxes with lighter shades indicates that features strongly correlated with each other. Based on this plot features which were highly correlated with the target variable were only selected for further model building and prediction. So features like ‘Total_amount','Tolls_amount','Tip_amount','Trip_distance' were selected for further analysis.

Models Used for Prediction :

1. Linear Regression:

Linear Regression is used for predicting a certain numerical value based on one or more variables. This method assumes to have a linear relationship between the target variable and other variables

- Mathematical Representation :
  y = b0 + b1x1 + b2x2 + ……. + bnxn + e
  where y : target variable to be predicted
  x1, x2 , xn … : are predictor variables
  b0 : is the intercept value
  b1 , b2 , bn … : are co efficients
  e : error term
- Key Features:
  1. Easy to implement and interpret.
  2. Assumes linear relationship between features and target
- Advantages :
  1. Computationally efficient.
  2. Works well with large datasets where the relationship is approximately linear.
- Disadvantages :
  1. Assumes linearity, which may not always be the case.
  2. Sensitive to outliers.
  3. Limited in capturing complex relationships.

2. Decision Tree Regressor:

Decision Tree Regression is a non-parametric supervised learning method used for both classification and regression tasks. It forms a tree like structure by splitting data into subsets based on input values of features

- Mathematical Representation:
  This model is represented as a series of if-else conditions
- Key Features :
  1. Easy to visualize and interpret.
  2. No need for feature scaling.
- Advantages :
  1. Handles non-linear relationships well.
  2. Robust to outliers (depending on the depth of the tree).
- Disadvantages :
  1. Prone to overfitting, especially with deep trees.
  2. Can be unstable as small changes in the data can result in a different tree.
  3. Less interpretable as trees grow deeper.

3. Random Forest regressor:

Random Forest Regressor is an ensemble method that builds multiple decision trees and merges their results to improve accuracy and control overfitting. It uses bagging (bootstrap aggregating) to create multiple subsets of data for training each tree.

- Mathematical Representation :
  This model prediction is the average of predictions from all individual trees.
- Key Features :
  1. Inherently performs feature selection.
  2. Captures complex relationships and interactions between features.
- Advantages :
  1. Robust to overfitting due to averaging.
  2. Handles large datasets well.
  3. Can handle missing values and maintain accuracy.
- Disadvantages :
  1. More computationally intensive than individual decision trees.
  2. Less interpretable compared to a single decision tree.
  3. Requires careful tuning of hyperparameters.

Evaluation Metrics for all the 3 ML models :

The problem statement is about predicting a numerical value and so the evaluation metrics considered as Mean squared error and R - square values

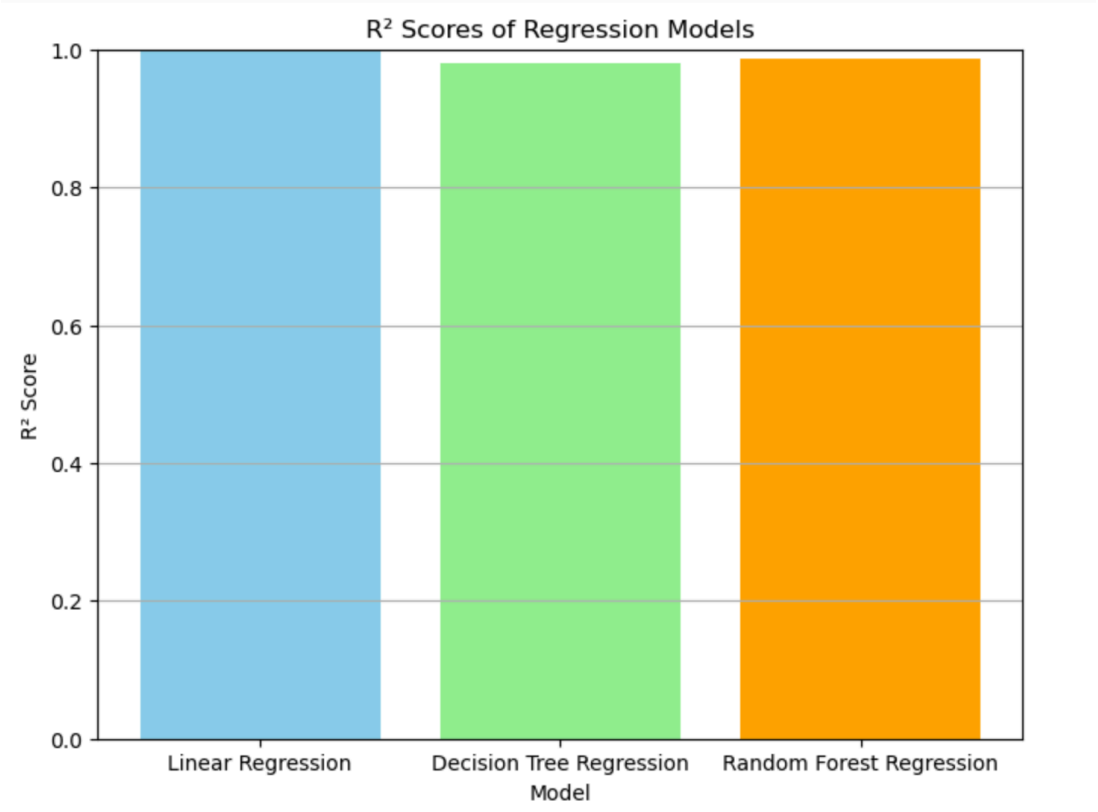
Linear Regression :
MSE : 0.0010583515060298497
R- Squared : 0.998049054435737

Decision Tree Regressor :
MSE : 0.010523161154485141
R- Squared : 0.9806017996295184

Random Forest Regressor :
MSE : 0.006828708679419709
R- Squared : 0.9874120848963174

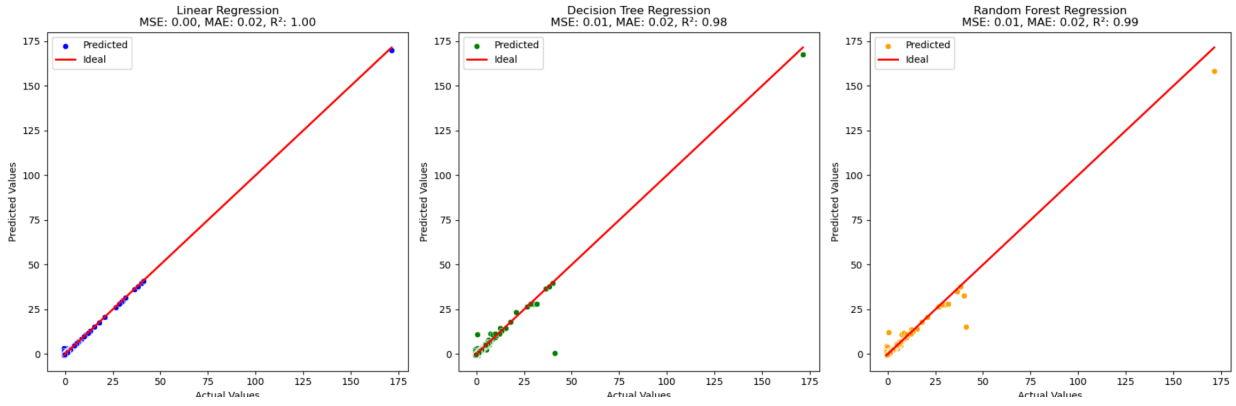
Model Comparison

The bar chart below compares the R- squared values for all the three models. From the plot it can be interpreted that all of the three models have pretty good and fair R squared values . Stating that they are more general to the unseen data. Linear Regression outperforms all of the three and is more generalizable model. So the best model for this data is Linear regression.



Based on the the plot and the evaluation metrics it can be concluded that all 3 models have very low MSE and high R- square values.

Actual and Predicted Values on Test Data :



From the above plot it can be observed that Red line are the actual values and Blue dots are the predicted values. The first plot is for the predicted values using Linear regression model and it can be interpreted that they pretty well line on the line and are very close to actual values. For

Decision tree Regressor the predicted values are a bit far from the actual values and for the Random Forest Regressor as well some of the points are diverted far from the actual values.

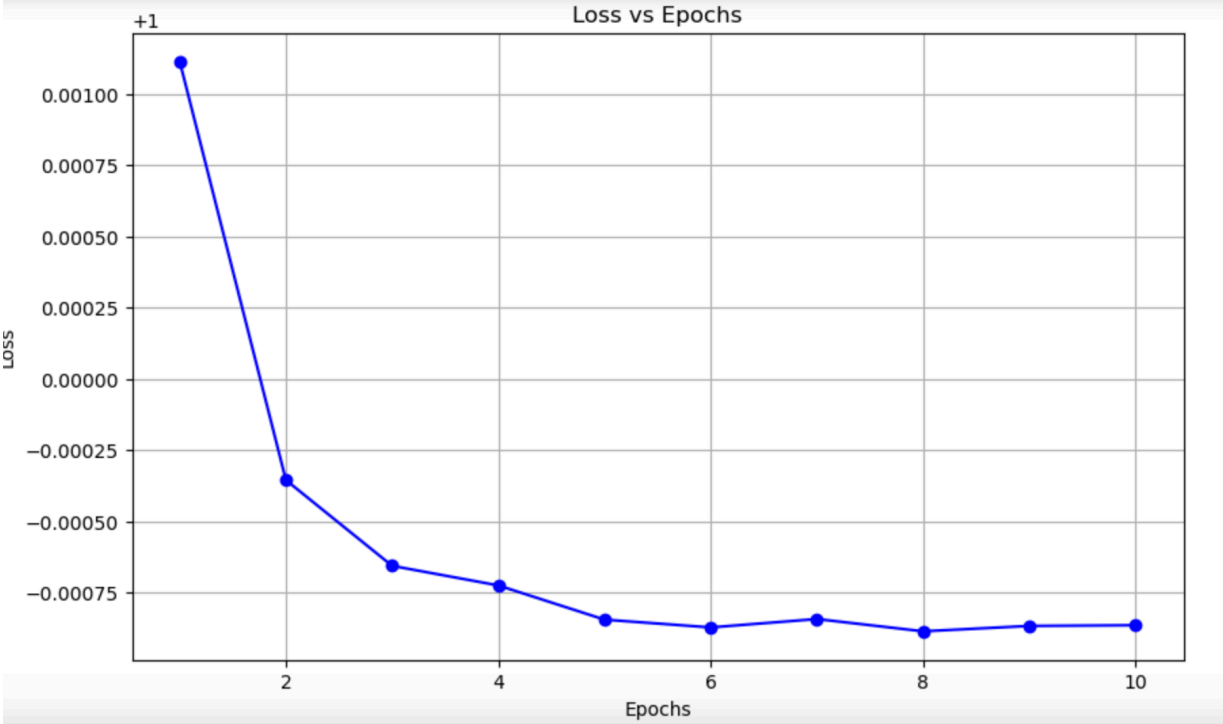
Neural Network Model for Problem Statement :

The neural network structure is a simple feedforward neural network (also known as a multilayer perceptron) for a regression problem (predicting a value).

The detailed structure of the Neural Network setup Used is as follows :

1. Input Layer: The input layer consists of 4 features.
2. Hidden Layer: There is one hidden layer with 50 neurons. The activation function used for this layer is the ReLU (Rectified Linear Unit) function.
3. Output Layer: The output layer consists of a single neuron, which is typical for regression problems where the output is a single continuous value.
4. Loss Function: The Mean Squared Error (MSE) loss function is used because this is a regression problem.
5. Optimizer: The Adam optimizer is used with a learning rate of 0.001.

Plot of Training Loss over the time (epochs) :



The plot indicated a decreasing trend in the training loss over the epochs. This shows that the model is learning well from the training data. And also there is some scope to check further epochs as there is no drastic difference between the loss after the 8th epoch.

PART III : OCTMNIST Classification

Neural Network Structure Used :

The neural network setup is CNN network designed for image classification . the detailed structure of the the set up is as follows :

Convolutional Layers: 2 layers with ReLU activations and max-pooling.
1. The **first layer** performs convolution with 32 filters, each of size 3x3 and has a padding of 1. Activation Function used is ReLU (Rectified Linear Unit) and is followed by Max Pooling with a 2x2 window, reducing the spatial dimensions .
2. The second layer performs convolution with 64 filters, each of size 3x3 and has a padding of 1 . Activation Function used ReLU applied after convolution and is followed by Max Pooling with a 2x2 window, further reducing the spatial dimensions.

Fully Connected Layers: 2 layers, with ReLU activation in the first.
1. The first layer has 128 neurons, each fully connected to the input features. Activation Function used is ReLU applied after the linear transformation.
2. The second layer maps the 128 features to 4 output classes. No activation function is specified here, the raw scores are the output.

Output: 4 logits, representing the raw scores for 4 classes.

Techniques Used to Improve the architecture of the NN Setup :

1. **DroupOut** : Dropout has reduced overfitting by ensuring that the network does not become too reliant on specific neurons. Also it helped me achieve stable and high validation accuracy. Also it made my model more generalized as in epoch 8 there is a slight increase in validation loss and a drop in validation accuracy, but in epoch 9 and epoch 10 the model quickly recovers in the subsequent epochs, indicating that dropout has helped in maintaining generalization without significant overfitting.

2. **Early Stopping** : Early stopping has prevented the model from continuing to train after reaching an optimal point, avoiding unnecessary epochs that could lead to overfitting.Also it has helped to save from the epoch with the best validation performance, ensuring that the best-performing model is used. in epoch 10 the state where validation performance was best,

as early stopping prevented the model from degrading due to overtraining. This tells that the final reported accuracies and losses are optimal.

Results of Model Performances :

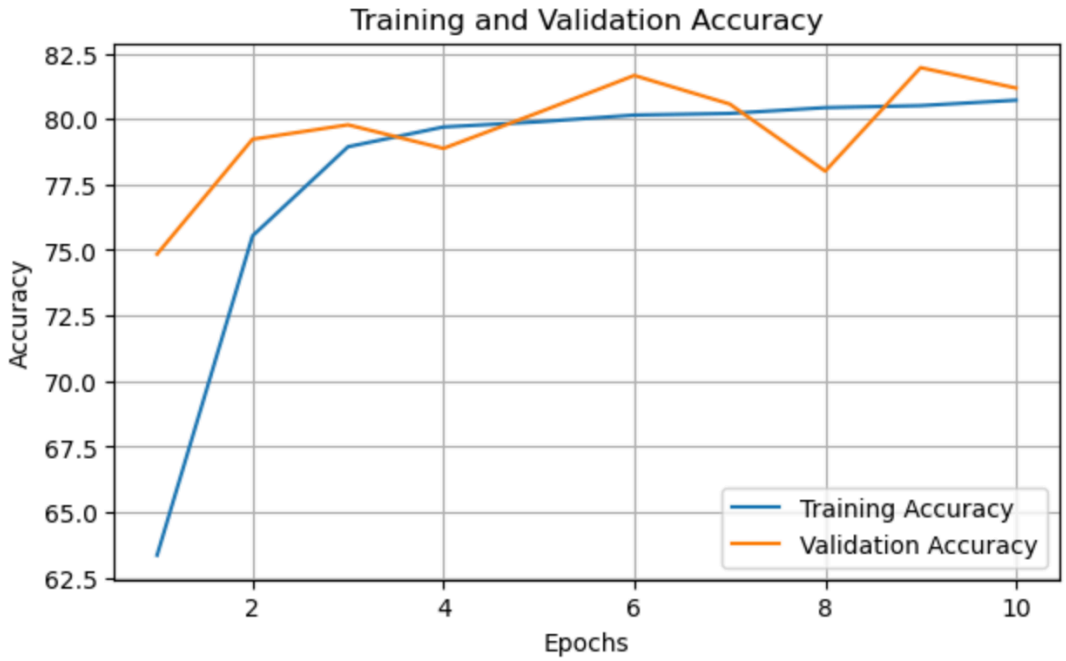
1. **Training accuracy, training loss, validation accuracy, validation loss .**

```
Epoch 1, Training Loss: 0.9387, Validation Loss: 0.7174, Training Accuracy: 63.36%, Validation Accuracy: 74.84%
Epoch 2, Training Loss: 0.7021, Validation Loss: 0.6068, Training Accuracy: 75.54%, Validation Accuracy: 79.23%
Epoch 3, Training Loss: 0.6177, Validation Loss: 0.5882, Training Accuracy: 78.94%, Validation Accuracy: 79.77%
Epoch 4, Training Loss: 0.5973, Validation Loss: 0.5881, Training Accuracy: 79.69%, Validation Accuracy: 78.87%
Epoch 5, Training Loss: 0.5897, Validation Loss: 0.5750, Training Accuracy: 79.89%, Validation Accuracy: 80.25%
Epoch 6, Training Loss: 0.5833, Validation Loss: 0.5569, Training Accuracy: 80.15%, Validation Accuracy: 81.66%
Epoch 7, Training Loss: 0.5804, Validation Loss: 0.5577, Training Accuracy: 80.21%, Validation Accuracy: 80.57%
Epoch 8, Training Loss: 0.5753, Validation Loss: 0.6260, Training Accuracy: 80.43%, Validation Accuracy: 78.01%
Epoch 9, Training Loss: 0.5737, Validation Loss: 0.5428, Training Accuracy: 80.51%, Validation Accuracy: 81.95%
Epoch 10, Training Loss: 0.5697, Validation Loss: 0.5394, Training Accuracy: 80.71%, Validation Accuracy: 81.18%
```

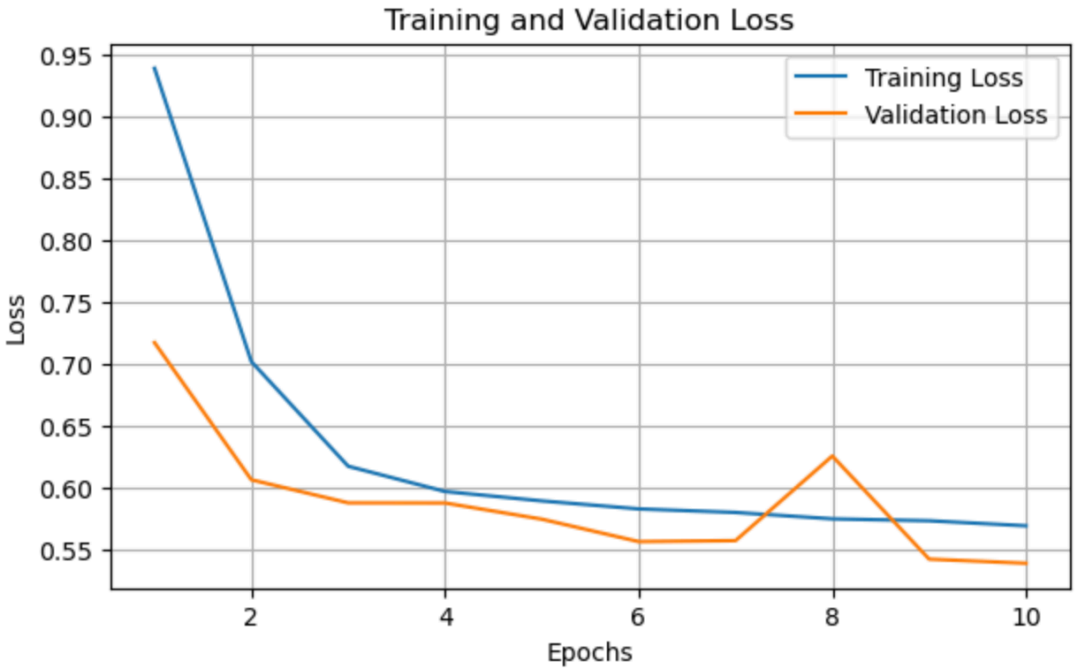
The model's performance appeared to be improving over the training epochs. The validation loss initially decreased from 0.7174 to 0.5394 at epoch 10. This may be because of DroupOut technique as the model is generalizing better on unseen data as the training progresses.

2. **Plot of training and validation accuracies over time (epochs)**

The training accuracy is higher than the Validation accuracy. Both of them are increasing indicating that model is learning well from the training data. Also the gap between both the curves is decreasing meaning model is working well on unseen data as well. There is some kind of constant pattern for training accuracy after epoch 6 but there is some variation in validation accuracy.

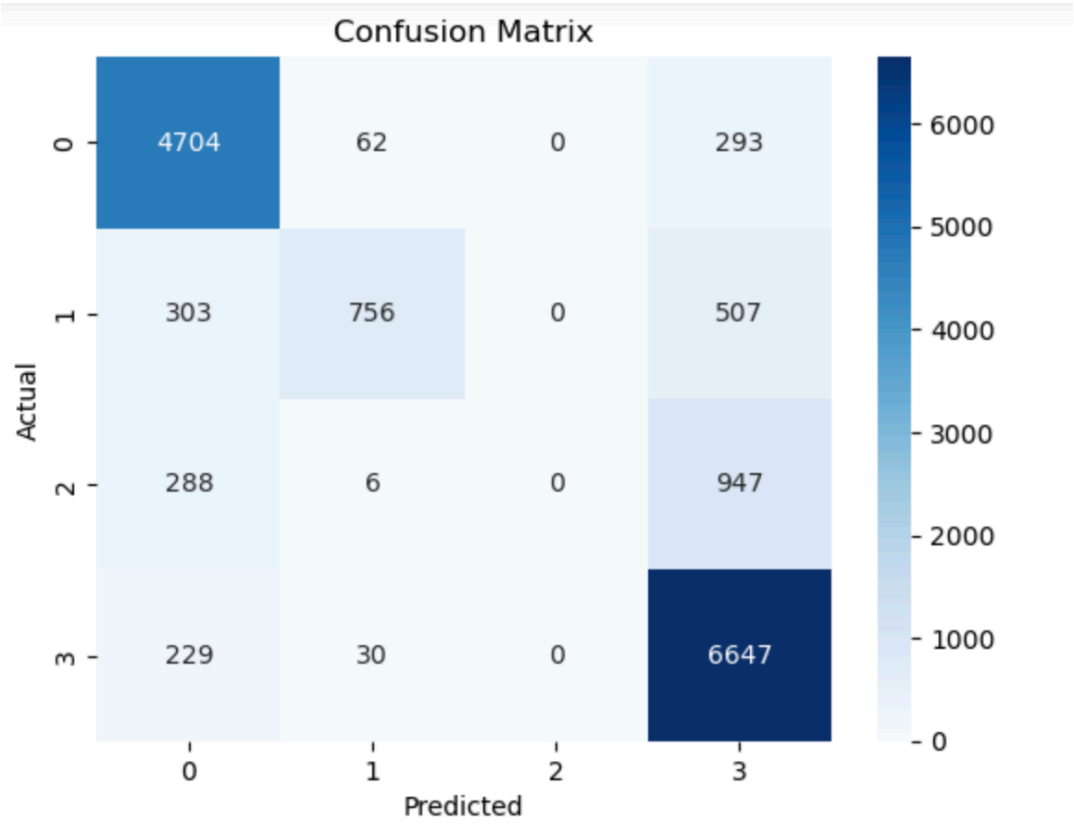


3. **Plot of training and validation loss over time (epochs)**



There is a significant decreasing trend in the training loss. Which indicates that the model is learning well. But after epoch 6 to epoch 10 the training loss is within a very specific and compact range. This shows there is some scope for the model to improve.

4. **Confusion matrix on test data**



The values on the diagonals indicate the number of correctly classified classes in the dataset. The values off the diagonal represent the misclassified classes. Although large number of sample are classified correctly but there are still significant number of samples that are misclassified

5. Precision , Recall and F1-Score of Prediction on test data:

```
precision, recall, f1, _ = precision_recall_fscore_support(all_labels, all_preds, average='weighted')

print(f'Precision: {precision:.4f}, Recall: {recall:.4f}, F1 Score: {f1:.4f}')
```

```
Precision: 0.7557, Recall: 0.8196, F1 Score: 0.7769
```

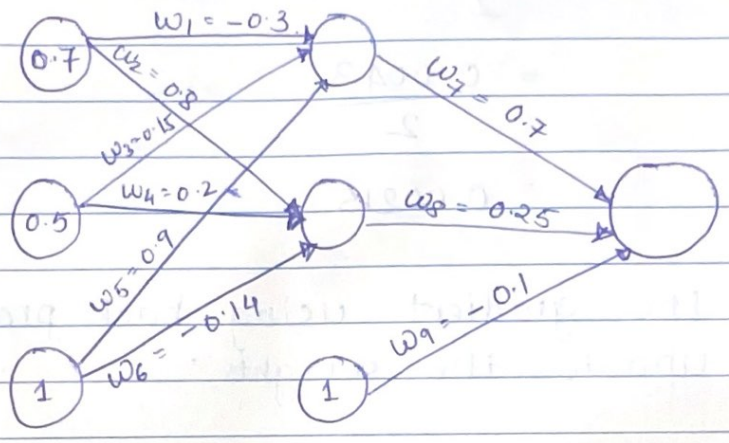
- Precision: Precision measures the fraction of positive predictions that are actually correct. Precision is 0.7557 this indicates that out of all the instances the model predicted as positive, 75.57% were actually positive.
- Recall: Recall measures the fraction of positive instances that were correctly identified by the model.recall is 0.8196 this indicates that out of all the actual positive instances in the data, the model identified 81.96% of them correctly.
- The precision and recall values are fairly close, which suggests a balanced classification task. The model is performing reasonably well on both identifying true positives and avoiding false positives.

Date: / /

PART II : Deep Learning Theoretical Part

1. Forward- Backward Pass.

Hidden layer activation: ReLU
Output layer activation: Linear
$\alpha = 0.03$; $y = 0.5$



1. Perform a forward pass and estimate predicted output

⇒ $h_1 = i_1 w_1 + i_2 w_3 + w_5$
$= (0.7)(-0.3) + (0.5)(0.15) + 0.9 = 0.765$.

h_1 Passing in activation function (ReLU)
⇒ $h_1 = 0.765$ as $x > 0$.

$h_2 = i_1 w_2 + i_2 w_4 + w_6$
$= (0.7)(0.8) + (0.5)(0.2) + (-0.14) = 0.52$

Passing in activation function (ReLU)
⇒ $h_2 = 0.52$.

Output $= h_1 w_7 + h_2 w_8 + w_9$
$= 0.765 \times 0.7 + 0.52\,(0.25) + (-0.1) = 0.5655$

Date: / /

Passing through activation function.

$\Rightarrow$ Output $= 0.5655$.

2. Estimate the MSE

$\rightarrow$ Calculate the error function using formula: $\text{Error} = \frac{1}{2}(y-\hat{y})^2$

$y = 0.5$ and $\hat{y} = 0.5655$

$\therefore$ MSE $= \frac{1}{2}(0.5-0.5655)^2$

$= \frac{0.0043}{2}$

$= 0.00215$.

3. Find the gradient using back propogation
4. and update the weights.

$\rightarrow$ start with w_7;

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

$$\left(\frac{\partial \text{Error}}{\partial w_7}\right) = \frac{\partial \text{Error}}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial w_7}$$

$\text{Error} = \frac{1}{2}(y-\hat{y})^2$ and $\hat{y} = h_1 w_7 + h_2 w_8 + w_9$.

$$\therefore \left(\frac{\partial \text{Error}}{\partial w_7}\right) = \frac{\frac{1}{2}(y-\hat{y})^2}{\partial \hat{y}} \times \frac{(h_1 w_7 + h_2 w_8 + w_1)}{\partial w_7}$$

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

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

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

$$= 0.6985.$$

Date: / /

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

$\frac{\partial \text{Error}}{\partial w_8} = -(y-\hat{y})h_2$.

$\therefore w_8 = 0.25 - (0.03)(-(0.0655) \times 0.752)$

$= 0.2490$.

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

$\frac{\partial \text{Error}}{\partial w_9} = -(y-\hat{y})$

$\therefore w_9 = -0.1 - (0.03)(-(0.0655))$.

$= -0.1020$.

For w_1; $w_1 = w_1 - \alpha\left(\frac{\partial \text{Error}}{\partial w_1}\right)$

$\therefore \frac{\partial \text{Error}}{\partial w_1} = \frac{\partial \text{Error}}{\partial \hat{y}} \times \frac{\partial \hat{y}}{\partial h_1} \times \frac{\partial h_1}{\partial w_1}$.

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

$= -(y-\hat{y}) \times w_7 \times i_1$.

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

$= -0.301$.

Similarly $w_2 = w_2 - \alpha\left(\frac{\partial \text{Error}}{\partial w_2}\right)$

$\frac{\partial \text{Error}}{\partial w_2} = -(y-\hat{y}) \times w_8 \times i_1$

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

$= 0.8 + 0.00034 = 0.8003$.

Date

$$w_3 = w_3 - \alpha\left(\frac{\partial Error}{\partial w_3}\right)$$

$$\frac{\partial Error}{\partial w_3} = -(y-\hat{y}) \times w_7 \times I_2$$

$$\therefore w_3 = 0.15 - (0.03)\left(-(0.0655) \times 0.7 \times 0.5\right)$$
$$= 0.15 + 0.00069.$$
$$= 0.1507.$$

$$w_4 = w_4 - \alpha\left(\frac{\partial Error}{\partial w_4}\right)$$

$$\frac{\partial Error}{\partial w_4} = -(y-\hat{y}) \times w_8 \times I_2.$$

$$\therefore w_4 = 0.2 - (0.03)\left(-(0.0655) \times 0.25 \times 0.5\right).$$
$$= 0.2 + 0.00025$$
$$= 0.2003.$$

$$w_5 = w_5 - \alpha\left(\frac{\partial Error}{\partial w_5}\right).$$

$$\frac{\partial Error}{\partial w_5} = -(y-\hat{y}) \times w_7.$$

$$\therefore w_5 = 0.9 - (0.03)\left(-(0.0655) \times 0.7\right).$$
$$= 0.9 + 0.0014.$$
$$= 0.9014.$$

$$w_6 = w_6 - \alpha\left(\frac{\partial Error}{\partial w_6}\right)$$

$$\frac{\partial Error}{\partial w_6} = -(y-\hat{y})\, w_8.$$

Date: / /

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

$= -0.14 + 0.00049$

$= -0.1395$

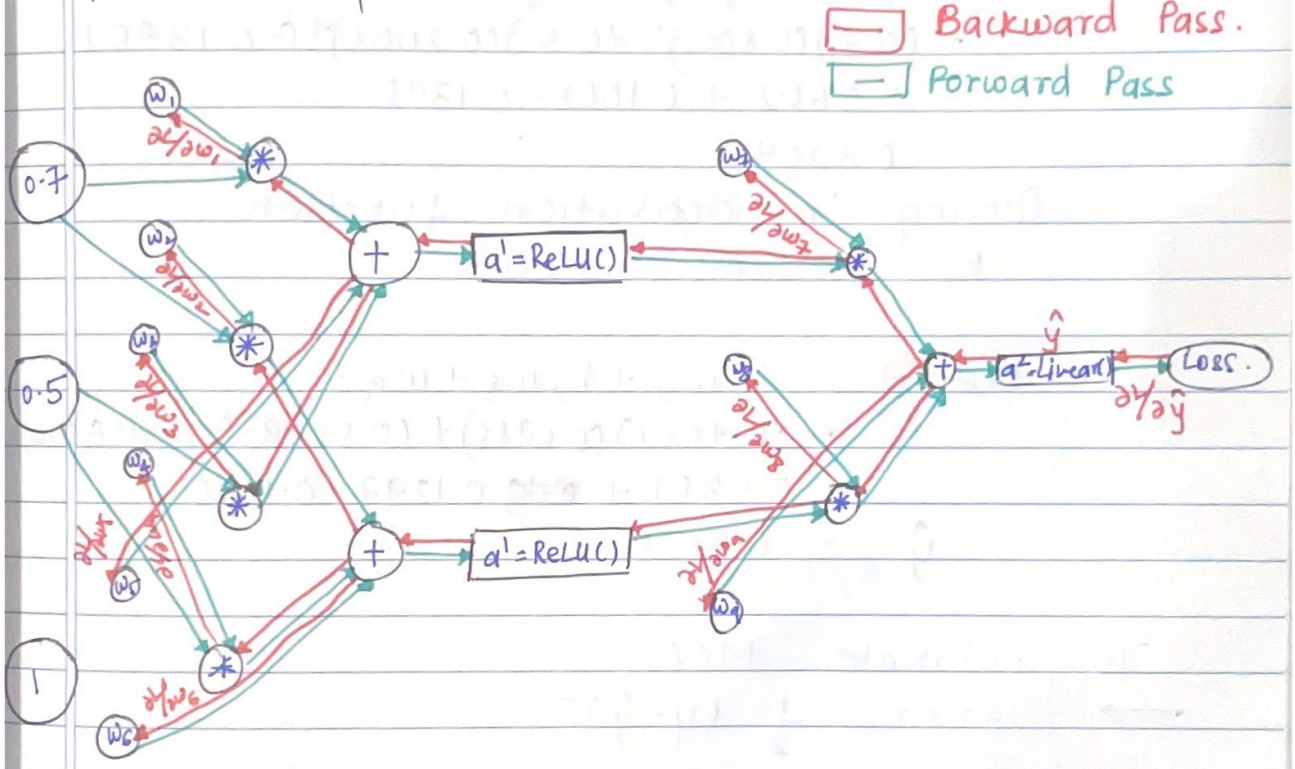
Thus updated weights are;

$w_1 = -0.301$, $w_2 = 0.8003$, $w_3 = 0.1507$, $w_4 = 0.2003$

$w_5 = 0.9014$, $w_6 = -0.1395$, $w_7 = 0.6985$,

$w_8 = 0.2490$, $w_9 = -0.1020$.

5. Draw a computation graph for forward and backward pass.



Date

6. Perform forward pass to estimate output using updated weights.

→ $h_1 = i_1 w_1 + i_2 w_3 + w_5$

$= (0.7)(-0.301) + (0.5)(0.1507) + 0.9014$

$= -0.2107 + 0.07535 + 0.9014$

$= 0.7661$

Passing in activation function

$h_1 = 0.7661 \quad x > 0$

$h_2 = i_1 w_2 + i_2 w_4 + w_6$

$= (0.7)(0.8003) + (0.5)(0.2003) + (-0.1395)$

$= 0.5602 + 0.1002 - 0.1395$

$= 0.5209$

Passing in activation function

$h_2 = 0.5209 \quad x > 0$

Output $= h_1 w_7 + h_2 w_8 + w_9$

$= (0.7661)(0.6985) + (0.5209)(0.2490) + (-0.10$

$= 0.5351 + 0.1297 - 0.1020$

$\hat{y} = 0.5628$

7. Estimate MSE

→ Error $= \frac{1}{2}(y - \hat{y})^2$

$= \frac{1}{2}(0.5 - 0.5628)^2$

$= 0.00197$

Thus the error has been reduced after updating the weights.

Date / /

Derivative of Tanh [5 points]

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Prove $f'(x) = 1 - f(x)^2$.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

We know;

$$f'\left(\frac{u}{v}\right) = \frac{u'v - v'u}{v^2}$$

So here $u = e^x - e^{-x} \Rightarrow u' = e^x + e^{-x}$

$v = e^x + e^{-x} \Rightarrow v' = e^x - e^{-x}$

$$\therefore f'(x) = \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2}$$

$$= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

$$= \frac{(e^x + e^{-x})^2}{(e^x + e^{-x})^2} - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

$$= 1 - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2}$$

$$= 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^2$$

and $\frac{e^x - e^{-x}}{e^x + e^{-x}} = f(x)$

$$\therefore f'(x) = 1 - f(x)^2$$

Hence proved.