Department of Electronic and Telecommunication Engineering

University Of Moratuwa

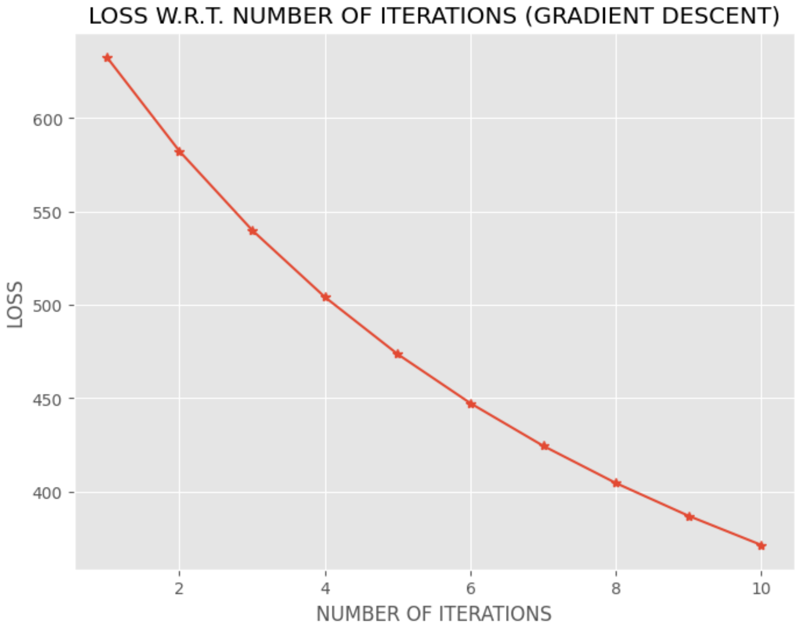
Assignment 02: Learning from data and related challenges and classification

Index No.	Name
200702H	P.D.G.U.M.B. WEERASINGHE

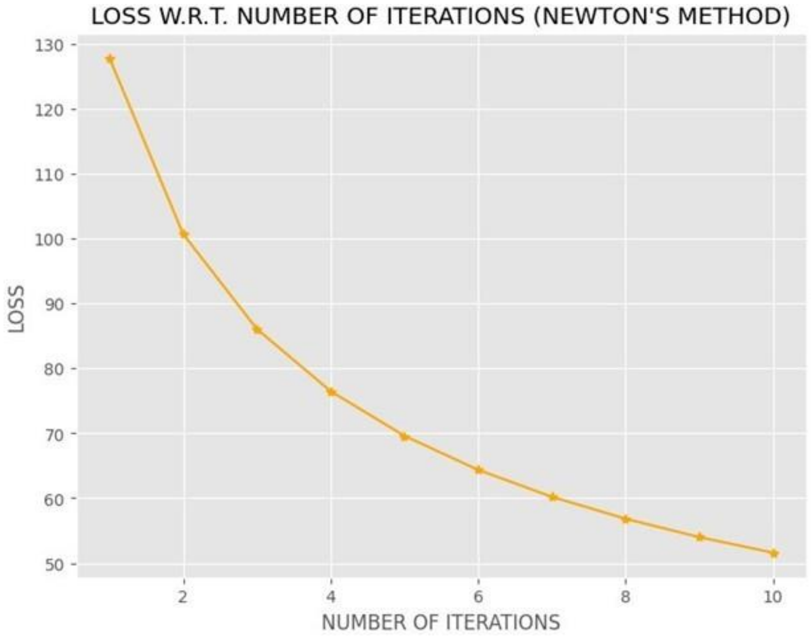
This assignment is submitted as a partial fulfillment of the module EN3150 – Pattern Recognition

1. Logistic regression weight update process

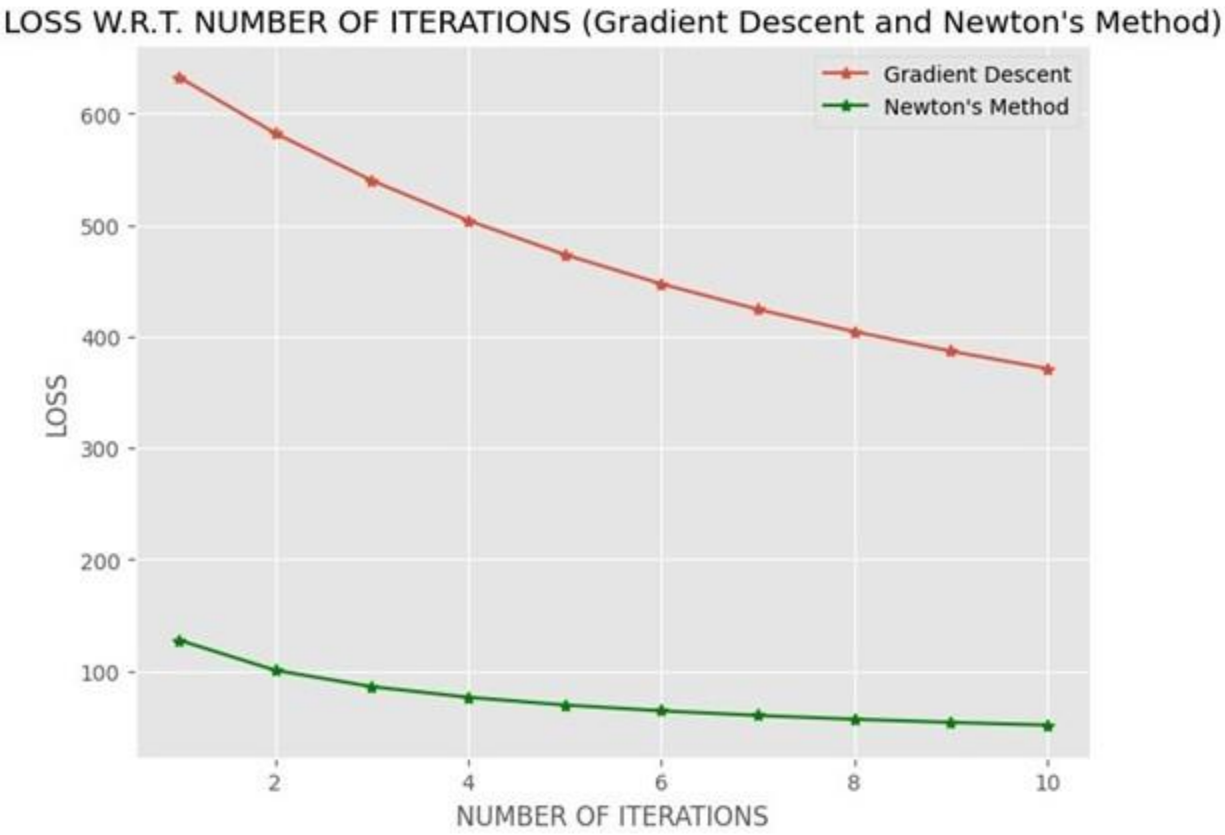
iii.



v.



vi.



Comments.

- Gradient Descent and Newton's Method are both used for logistic regression with binary cross-entropy loss.
- Gradient Descent updates the weights using the gradient of the loss, while Newton's Method uses the Hessian matrix.
- Although Newton's Method may require more computing, it often converges more quickly.
- For both strategies, the displayed graph demonstrates that the loss decreases with each repetition.
- In this situation, Newton's Method most likely converges more quickly and achieves a lower loss value in fewer rounds.

Discussion

```
W values | 10 iterations later | Gradient Descent:
W0: 0.009031764884538475
W1: 0.262301155857097
W2: 0.4994938404035172
```

```
W Values | 10 iterations later | Newton's Method :
W0: 3.145757812061996
W1: 2.9679207749775514
W2: 1.4806766332837635
```

We can see that after 10 iterations W values of Gradient descent are smaller than the w values of Newton’s method. It indicates that loss of the gradient descent is higher than the loss of the Newton’s method. So the convergence rate will be higher in newton’s method.

```
Iteration | Gradient Descent Loss | Newton's Method Loss
--------------------------------------------------------
   1     |       632.8212         |       127.8597
   2     |       582.3973         |       100.7177
   3     |       540.0400         |       86.0785
   4     |       504.2051         |       76.5029
   5     |       473.6390         |       69.6205
   6     |       447.3433         |       64.3793
   7     |       424.5299         |       60.2261
   8     |       404.5770         |       56.8375
   9     |       386.9932         |       54.0096
  10     |       371.3875         |       51.6069
```

These are the y axis values for obtained graphs. Newton’s method has reached a lower value than the Gradient Descent method. It proves the convergence speed of Newton’s method. More iteration is required for the Gradient Descent method to reach this value which Newton Method has reached.

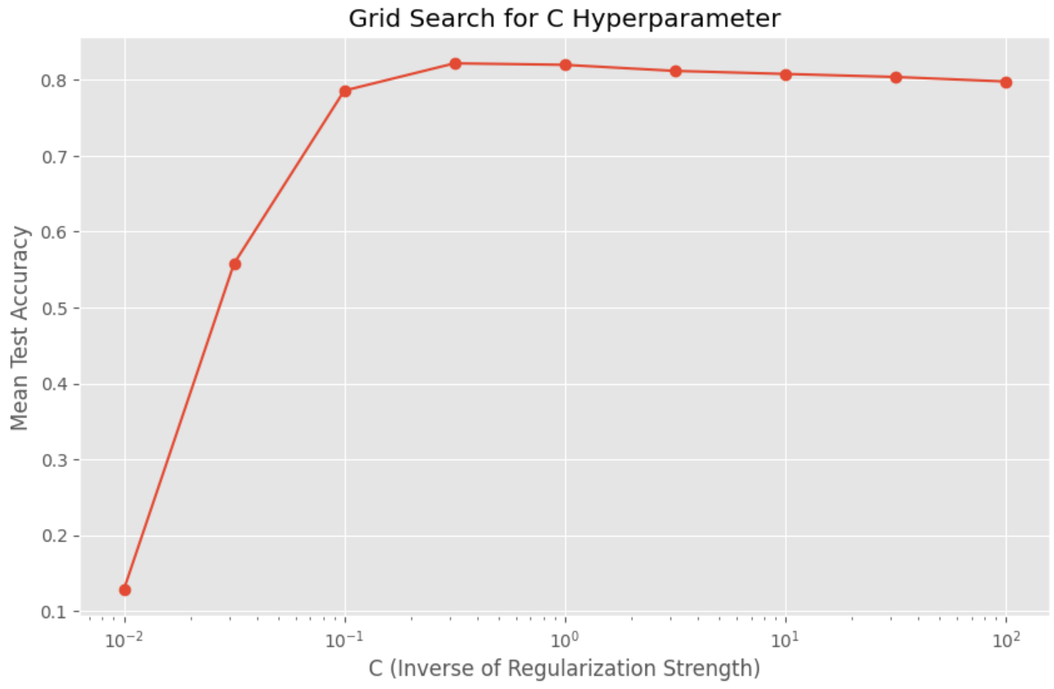
2. Perform grid search for hyper-parameter tuning.

ii.

- In the context of data preprocessing in machine learning, the dataset undergoes a shuffling process, facilitated by the employment of the expressions "X = X[permutation]" and "Y = Y[permutation]."
- The pivotal component of this code is the variable denoted as "permutation," which signifies an array of indices meticulously constructed to embody a random rearrangement of the dataset's original indices.
- This permutation operation is systematically applied to both the feature matrix denoted as 'X' and the target variable, denoted as 'Y.'
- The objective of this shuffling process is twofold: firstly, it serves to eliminate any potential biases that may be inherent in the initial organization of the data.

Secondly, it ensures that the data employed for both training and testing purposes is entirely randomized in its distribution, thereby enhancing the robustness and reliability of subsequent machine learning analyses.

V.

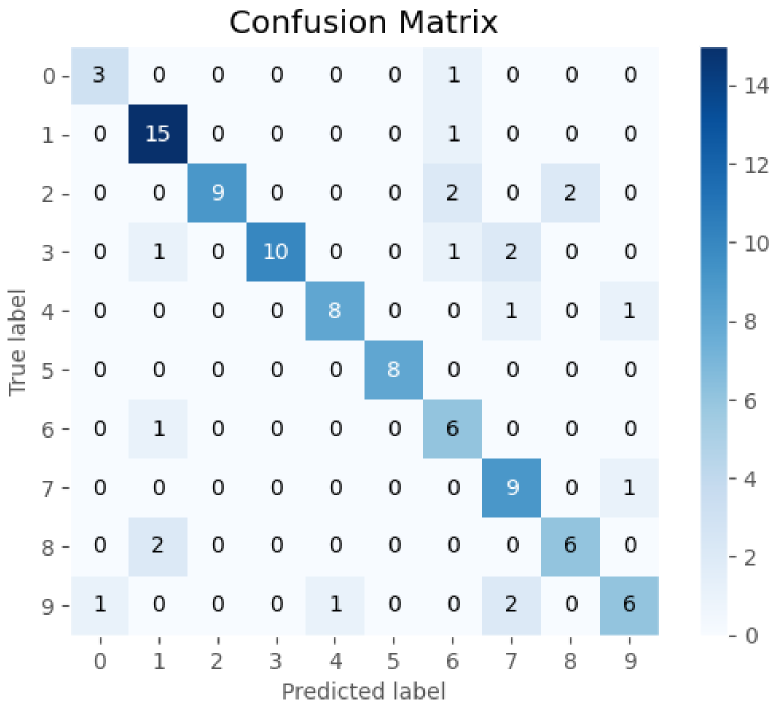


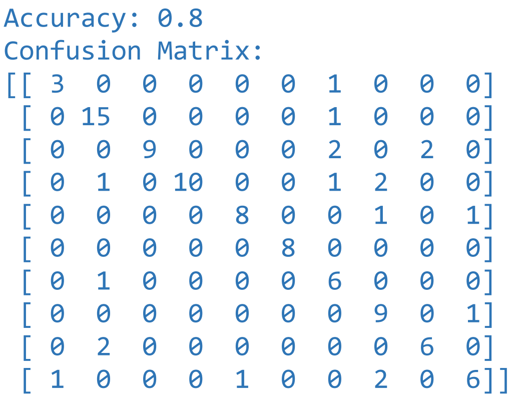
The presented graphical representation describes the relationship between the mean test score, denoting classification accuracy, and the hyperparameter C, which exhibits variations on a logarithmic scale. The following observations and insights can be discerned from the plot:

1. **Increasing C Enhances Test Score:** As the hyperparameter C ascends from 0.01 to 1.0, a discernible trend emerges wherein the mean test score exhibits a gradual and consistent improvement. This pattern suggests that decreasing the strength of regularization, which is associated with higher C values, results in a superior model performance on the test dataset.
2. **Optimal C Value at Approximately 1.0:** Evidently, the plot highlights a peak in the mean test score at a hyperparameter value of approximately 1.0, corresponding to a mean test score of approximately 0.83. This peak signifies the point at which the model attains its highest level of performance, implying that a moderate level of regularization is most efficacious for this specific dataset and model pairing.

3. **Diminishing Test Score Beyond C=1.0:** Beyond the optimal value of C, namely C=1.0, a subtle decline in the mean test score becomes discernible. This trend implies that excessive reduction in regularization, potentially leading to overfitting, results in a decline in model performance on the test dataset.
4. **Visualizing the Regularization-Performance Trade-off:** Overall, the graphical representation serves as an invaluable tool for comprehending the intricate interplay between the regularization strength denoted by C and the model's performance. It underscores the imperative nature of selecting an appropriate C value, one that adeptly balances the inherent trade-off between bias and variance in the model, ultimately culminating in optimal generalization capabilities for unforeseen data. In the context of this particular analysis, the hyperparameter C set at 1.0 emerges as a judicious choice, striking a harmonious equilibrium between regularization and model performance.

vi.





```
Accuracy: 0.8
Confusion Matrix:
[[ 3  0  0  0  0  0  1  0  0  0]
 [ 0 15  0  0  0  0  1  0  0  0]
 [ 0  0  9  0  0  0  2  0  2  0]
 [ 0  1  0 10  0  0  1  2  0  0]
 [ 0  0  0  0  8  0  0  1  0  1]
 [ 0  0  0  0  0  8  0  0  0  0]
 [ 0  1  0  0  0  0  6  0  0  0]
 [ 0  0  0  0  0  0  0  9  0  1]
 [ 0  2  0  0  0  0  0  0  6  0]
 [ 1  0  0  0  1  0  0  2  0  6]]
```

```
Precision: 0.8326722108301056
Recall: 0.8

F1 Score: 0.802717019822283
```

In assessing the performance of the model under consideration, several noteworthy observations emerge, as follows:

1. **Accuracy (0.8):** The model's accuracy stands at 0.8, signifying that it correctly classified 80% of the test samples. While accuracy offers a valuable overall assessment of model performance, it may not provide a comprehensive understanding, particularly in scenarios involving imbalanced datasets.
2. **Confusion Matrix:** The confusion matrix serves as a pivotal tool for a granular evaluation of the model's predictions vis-à-vis the true class labels:
    - Diagonal Elements: The diagonal elements, spanning from the top-left to the bottom-right, denote the number of correct predictions for each class.
    - Off-diagonal Elements: Conversely, the off-diagonal elements signify instances of misclassification. For example, the entry at (0, 0) reveals that three samples from class 0 were accurately classified, while the value at (2, 4) indicates that two instances from class 2 were erroneously classified as class 4.
3. **Precision (0.8327):** Precision, measuring the proportion of predicted positive instances that were indeed positive, yields a weighted precision score of 0.8327. This implies that, on average, 83.27% of positive predictions made by the model were correct.
4. **Recall (0.8):** Recall, also known as sensitivity or the true positive rate, gauges the percentage of actual positive instances that were accurately predicted. With a recall value of 0.8, the model successfully identified 80% of the actual positive instances.
5. **F1 Score (0.8027):** The F1 score, a harmonious amalgamation of precision and recall, artfully balances the interplay between these two performance indicators. An F1 score of 0.8027 denotes a reasonably well-balanced trade-off between precision and recall, attesting to the model's effectiveness.
6. **Balanced Trade-off with F1-score:** The F1-score, calculated at approximately 0.8, signifies a reasonably balanced equilibrium between precision and recall across all classes. This metric underscores the model's capacity to harmonize the trade-off between these crucial performance indicators.
7. **Potential for Improvement:** To further enhance the model's performance, several strategies can be explored. Firstly, a meticulous examination of hyperparameters is recommended, which may entail fine-tuning to optimize model behavior. Secondly, addressing class imbalance, if it prevails, can ameliorate the model's ability to handle different categories effectively. In sum, while the model exhibits promise, it is likely that judicious fine-tuning and data preprocessing measures may be employed to unlock its full classification potential.

In summation, the model's performance appears commendable, as indicated by its high accuracy and balanced F1 score. However, it is paramount to contextualize these metrics within the specific needs of the application. Furthermore, conducting in-depth analyses, such as scrutinizing misclassified samples or exploring domain-specific metrics, can offer deeper insights into the model's functionality and uncover avenues for potential enhancement.

3 Logistic regression

a)

$$P(Y=1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

$$P(Y=1) = \frac{1}{1+e^{-(-6+0.05.40+1\times3.5)}}$$

$$P(Y = 1) = 0.37754$$

$$P(Y=1) \approx 0.3775$$

With 40 hours of study and a 3.5 undergraduate GPA, the probability that a student will earn an A+ in the class is therefore predicted to be around 37.754%.

b)

$$0.5 = \frac{1}{1 + e^{-(w0+w1.x1+w2.x2)}}$$

$$0.5 = \frac{1}{1 + e^{-(-6+ 0.05.x_1 +1\times3.5)}}$$

$$X_1 = 50$$

Therefore, a student like the one in component (a) needs to study for roughly 50 hours to have a 50% chance of getting an A+ in the class.